МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри_____ Є. О. Давиденко

підпис

«___» _____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Вебзастосунок для кафе навинос

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 408.22010807

Здобувачка _____А. В. Жирнова

підпис

«__» _____ 2024 р.

Керівник PhD, ст. викладач _____І. О. Кандиба

підпис

«__» _____ 2024 р.

Консультант канд. техн. наук, доцент _____А. О. Алексєєва

підпис

«__» _____ 2024 р.

м. Миколаїв – 2024 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Зав. кафедри Є. О. Давиденко

«23» лютого 2024 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи бакалавра

Видано здобувачці групи 408 факультету комп'ютерних наук

Жирновій Анастасії Віталівні

(прізвище, ім'я, по батькові здобувачки)

1. Тема кваліфікаційної роботи

Вебзастосунок для кафе навинос

Затверджена наказом по ЧНУ від «22» грудня 2023 р. № 269

2. Строк представлення кваліфікаційної роботи « _____» _____ 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом роботи є вебзастосунок для кафе навинос.

4. Перелік питань, що підлягають розробці: аналіз предметної області та аналогічних застосунків доставки їжі, визначення вимог та функціоналу застосунку, дизайн, проєктування бази даних, розробка ПЗ та тестування системи.

5. Перелік графічних матеріалів

Презентація.

6. Завдання до спеціальної частини

Охорона праці на робочих місцях фахівців в сфері обслуговування.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А. О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи PhD, ст. викладач Кандиба Ігор Олександрович

(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

Жирнова Анастасія Віталіївна

(прізвище, ім'я, по батькові здобувачки)

(підпис)

Дата видачі завдання «23» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН виконання
кваліфікаційної роботи

Тема: Вебзастосунок для кафе навинос

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	22.02.2024	23.02.2024	Виконано
2.	Огляд літератури за темою роботи	24.02.2024	26.02.2024	Виконано
3.	Складання календарного плану КРБ	26.02.2024	27.02.2024	Виконано
4.	Аналіз предметної області	28.02.2024	01.03.2024	Виконано
5.	Розробка проєктних рішень	04.03.2024	10.03.2024	Виконано
6.	Дизайн ПЗ	12.03.2024	20.03.2024	Виконано
7.	Моделювання та конструювання ПЗ	22.03.2024	26.03.2024	Виконано
8.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	28.03.2024	22.05.2024	Виконано
9.	Розробка спеціальної частини з охорони праці	29.05.2024	02.06.2024	Виконано
10.	Оформлення КРБ та презентації	19.04.2024	12.06.2024	Виконано
11.	Попередній захист	05.06.2024	05.06.2024	Виконано
12.	Завершення оформлення КРБ та презентації	12.06.2024	12.06.2024	Виконано
13.	Відгук керівника КРБ	14.06.2024	14.06.2024	Виконано
14.	Рецензування	14.06.2024	16.06.2024	Виконано
15.	Захист кваліфікаційної роботи	24.06.2024	27.06.2024	Виконано

Розробила здобувачка Жирнова А. В. _____
(*прізвище, ім'я, по батькові)* *(підпис)*

«27» лютого 2024 р.

Керівник роботи PhD, ст. викладач Кандиба І. О. _____
(*посада, прізвище, ім'я, по батькові)* *(підпис)*

«27» лютого 2024 р

АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Вебзастосунок для кафе навинос»

Здобувачка 408 гр.: Жирнова Анастасія Віталіївна

Керівник: PhD, ст. викладач Кандиба І. О.

Кваліфікаційна робота присвячена проєктуванню та розробці вебзастосунку кафе навинос. Актуальність теми обґрунтована тим, що цифровізація бізнесу набула значного розмаху, таким чином, концепція такого кафе орієнтується не лише на швидкість та автоматизацію, а й на захист клієнтів та дотримання норм безпеки сьогодення.

Об’єктом кваліфікаційної роботи є процес автоматизації замовлень, ведення обліку, та визначення потреб клієнта кафе навинос.

Предметом кваліфікаційної роботи є інструментарій розробки вебзастосунку кафе навинос.

Метою кваліфікаційної роботи є розробка вебзастосунку для покращення ефективності роботи кафе навинос.

Кваліфікаційна робота складається зі вступу, чотирьох розділів, висновків, переліку джерел посилань.

У вступі визначена актуальність теми роботи, предмет і об’єкт дослідження, та встановлено перелік завдань задля досягнення сформульованої мети.

У першому розділі проаналізовано існуючі аналоги, розглянуто специфіку. В результаті проведено огляд літератури та аналіз існуючих підходів до створення вебзастосунків.

У другому розділі здійснено моделювання ПЗ, розроблені проєктні рішення та сформовано функціональні вимоги до програмного забезпечення. Також описано інтерфейс користувача та сценарії використання.

У третьому розділі описано проєктування системи та технології, інструменти для розробки вебзастосунку. Також визначено чому обрані мови й технології відповідають потребам та цілям.

У четвертому розділі описано детальний процес розробки, також пояснення щодо реалізованого функціоналу для кожної ролі в системі – наведені результати програмної реалізації.

КРБ викладена на 61 сторінку, вона містить 4 розділи, 43 ілюстрації, 4 таблиці, 20 джерел в переліку посилань.

Ключові слова: *кафе навинос, вебзастосунок, Laravel, MySQL*

ABSTRACT

of the Bachelor's Thesis

«Web application for the cafe for takeaway»

Student of group 408: Zhyrnova Anastasiia Vitaliivna

Supervisor: PhD, Senior Lecturer Kandyba I.O.

The qualification work is devoted to the planning and development of a takeaway cafe web application. The relevance of the topic is justified by the fact that business digitalization has become widespread, so the concept of such a cafe focuses not only on speed and automation, but also on customer protection and compliance with current safety standards.

The **object of the qualification work** is the process of automating orders, keeping records, and determining the needs of the client of a takeaway cafe.

The **subject of the qualification work** is the tools for developing a web application for a takeaway cafe.

The **goal of the qualification work** is to develop a web application to improve the efficiency of a takeaway cafe.

The qualification work consists of an introduction, four chapters, conclusions and a list of references.

The introduction defines the relevance of the topic, the subject and object of the study, and sets out a list of tasks to achieve the stated goal.

In the first section, the existing analogs are analyzed, the specifics are considered. As a result, a literature review and analysis of existing approaches to creating web applications were conducted.

In the second section, the software is modeled, design solutions are developed, and functional requirements for the software are formed. The user interface and use cases are also described.

The third section describes the design of the system and technology, tools for developing a web application. Also, it also explains why the chosen languages and technologies meet the needs and goals.

The fourth section describes the detailed development process, explains the implemented functionality for each role in the system, and presents the results of the program implementation.

The bachelor's thesis consists of 61 pages, the work contains 4 chapters, 43 illustrations, 4 tables, 20 sources in the list of references.

Keywords: *cafe for takeaway, web application, Laravel, MySQL*

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ... 3
ВСТУП ... 4
1 АНАЛІЗ ВЕБЗАСТОСУНКІВ КАФЕ ... 6
1.1 Вплив технологій на сферу харчування та обслуговування ... 6
1.2 Дослідження вебзастосунків кафе ... 7
1.3 Специфікація вимог ... 12
Висновки до розділу 1 ... 16
2 ПРОЄКТУВАННЯ ВЕБСАЙТУ ДЛЯ КАФЕ НАВИНОС ... 17
2.1 Сценарії використання вебзастосунку ... 17
2.2 Розробка діаграми послідовностей ... 19
2.3 Розробка мокапів системи ... 21
2.4 Розробка моделі потоків даних системи ... 27
2.5 Розробка діаграми станів системи ... 28
Висновки до розділу 2 ... 30
3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ... 31
3.1 Опис технологій розробки ... 31
3.2 Розробка діаграми класів ... 34
3.3 Моделювання бази даних ... 35
Висновки до розділу 3 ... 43
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ ... 44
4.1 Реалізація вебсайту ... 44
4.2 Реалізація панелі адміністрування ... 47
4.3 Реалізація компонентів логіки застосунку ... 49
4.4 Тестування розробленого ПЗ ... 55
Висновки до розділу 4 ... 58
ВИСНОВКИ ... 59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ... 60

ПЕРЕЛІК СКОРОЧЕНЬ

CSS – Cascading Style Sheets

MVC – Model-View-Controller

ORM – Object-relational mapping

SQL – Structured Query Language

БД – база даних

КРБ – кваліфікаційна робота бакалавра

ПЗ – програмне забезпечення

ВСТУП

Цифровізація бізнесу в останні роки набрала великі оберти і кожен з власників женеться за увагою клієнта. Адже потрібно бути «на виду» – розвивати бізнес ще й онлайн через різні канали комунікації. Тому потрібно проявляти гнучкість в бізнесі, адже з початком пандемії, а потім й повномасштабного вторгнення умови безпеки та ваше місцезнаходження – є першочерговими пунктами щоденно. Отож, концепція кафе навинос – це не тільки про швидкість та автоматизацію процесів, а й про захист кожного та дотримання норм безпеки, аби уникати скупчення людей.

Сучасне життя кожного тим чи іншим шляхом, але зв’язане з технологіями, а отже у смартфоні – все, і він є нашим помічником в роботі, навчанні, там відбувається спілкування, можете закарбувати важливі моменти життя. Нажаль, це як й спрощує життя, економить час, та доступ до широкого вибору, але й це впливає на соціальність, відносини, перебування у великому потоці інформації та реалізація людини у суспільстві.

Сфера харчування завжди була та є найпопулярнішої серед усіх поколінь, тому розробка застосунка кафе навинос є актуальною, та важливими критеріями є врахування особливостей кафе навинос, можливість доставки, панель адміністратора для налаштувань меню. Вже й так багато хто перейшов на доставку продуктів з супермаркетів, через швидкість та небезпеку перебування в торговельних центрах, отож доставка з кафе стає ще одним зручним пунктом в рутинних справах. Але складність застосунку полягає в важливості врахування багатьох аспектів працездатності, безпеки даних та інтеграція, наприклад, зі складом запасів або бухгалтерський облік.

Об’єктом кваліфікаційної роботи є процес автоматизації замовлень, ведення обліку, та визначення потреб клієнта кафе навинос.

Предметом кваліфікаційної роботи є інструментарій розробки вебзастосунку кафе навинос.

Метою кваліфікаційної роботи є розробка вебзастосунку для покращення ефективності роботи кафе навинос

Для досягнення визначеної мети необхідно вирішити такі завдання:

1) аналіз предметної області;
2) визначення вимог до ПЗ;
3) визначення функціоналу усіх сторінок застосунку;
4) скласти сценарії використання;
5) створення дизайну вебзастосунку;
6) проєктування бази даних меню, користувачів;
7) розробка застосунку;
8) тестування системи.

1 АНАЛІЗ ВЕБЗАСТОСУНКІВ КАФЕ

1.1 Вплив технологій на сферу харчування та обслуговування

Жоден сучасний бізнес зможе існувати у довгостроковій перспективі без розробки стратегії цифрової трансформації. Зараз спостерігається стрімке прискорення тенденцій, що до сьогодні набирали популярність дуже повільно, саме вони визначать майбутнє бізнесу у наступні роки. Діджиталізація послуг, як і діджиталізація держави та технології в цілому допомогли сповільнити поширення кризи й дозволяють впоратися з новими реаліями[1].

Одним із ключових аспектів впровадження IT-технологій у галузі харчування є розробка мобільного застосунка для замовлення їжі. Користувачі можуть швидко та зручно вибирати страви, оплатити та отримати бажану їжу, вказавши адресу доставки – все це всього за декілька натискань по екрану. Це сприяє збільшенню зручності та швидкості обслуговування, а також дозволяє ресторанам збільшити свій обсяг продажів. Наприклад, заклади швидкого обслуговування, де ефективність є ключовим принципом, найбільш зацікавлені у впровадженні цифрових рішень.

Використання технологій у кафе відкриває величезні можливості. Ось деякі відмінні риси цього процесу:

– використання спеціалізованих програмних продуктів дозволяє кафе ефективно керувати замовленнями та інвентаризацією продукції. Це допомагає скоротити відходи, знизити витрати на керування та оптимізувати запаси;

– допомагає в управлінні через аналіз даних про замовлення, переваги клієнтів, обсяги продажу та інші параметри;

– цифрові меню дозволяють кафе швидко оновлювати свої акції та пропозиції.

Загалом цифрові системи дозволяють кафе розширювати свій бізнес, відкривати нові точки або розширювати асортимент товарів та послуг. Хоча цифровізація може принести багато переваг, важливо враховувати недоліки та

збалансувати їх для успішного впровадження у бізнесі. Але не можна забувати про такі важливі аспекти, як безпека та кібератаки – системи можуть бути вразливими, особливо платіжна інформація, і щоб залишатися на плаву і конкурентоспроможними, необхідно постійно оновлювати ПЗ.

1.2 Дослідження вебзастосунків кафе

Розгляд предметної області вебзастосунку для кафе на винос потребує дослідження різнопланових аспектів, таких як функціональність, продуктивність та зручність використання. Дуже важливо створити ПЗ, добре проаналізувавши аналоги та вимоги користувачів, адже від цього залежить й задоволеність клієнтів, й зрозумілість постановки задачі та визначення потенційних помилок роботи.

Сьогодні й так у нас є безліч вибору в усіх сферах, тому користувач є вимогливим. Загалом, для усіх онлайн ресторанів можна виділити головні переваги – доступність, швидкість та різноманітність вибору продуктів. Клієнти можуть легко обрати страви, додати їх у кошик та оформити замовлення за декілька кліків. Аналіз допоможе визначити сильні та слабкі сторони конкурентів, і розробити новий підхід для розвитку.

Огляд програмних рішень, які вже існують на ринку, дозволить зрозуміти переваги та недоліки різних підходів, функціональних вимог, дизайну з метою забезпечення більш ефективного рішення для потреб кафе навинос.

Для розробки застосунку розглянуто наступні аналоги, для повного розуміння необхідного функціонала:

- «Вилки-Палки»;
- «SimpleFood»;
- «NaDivanE».

«Вилки-Палки» – це програмне забезпечення, яке пропонує широкий спектр функцій для зручного замовлення та доставки їжі. Розроблене на базі фреймворку Laravel та фронтенду Vue.js, доступний для ПК, так й в мобільної версії [2].

Що стосовно функціоналу, то це можливість зареєструватися, далі меню з розділами, зручний пошук з фільтрами, а також робити передзамовлення. Наявність опції відстеження доставки забезпечує контроль над процесом. Зареєстровані клієнти можуть залишати відгуки.

Особистий кабінет включає в себе широкий функціонал, такий як статистика замовлень, відгуки і інформація по замовленням. Мапа з цінами доставки допомагає користувачам планувати свої замовлення.

Переваги:

– зручний інтерфейс;

– широкий функціонал, який включає в себе можливість реєстрації, фільтрації страв, відстеження доставки;

– система накопичення бонусів;

– можливість видалення інгредієнтів зі страв;

– можливість додавання улюблених страв в особистий кабінет, як зображено на рисунку 1.1.

Хоча недоліків не визначено критичних, можливість для подальшого розвитку та вдосконалення все ж таки є.

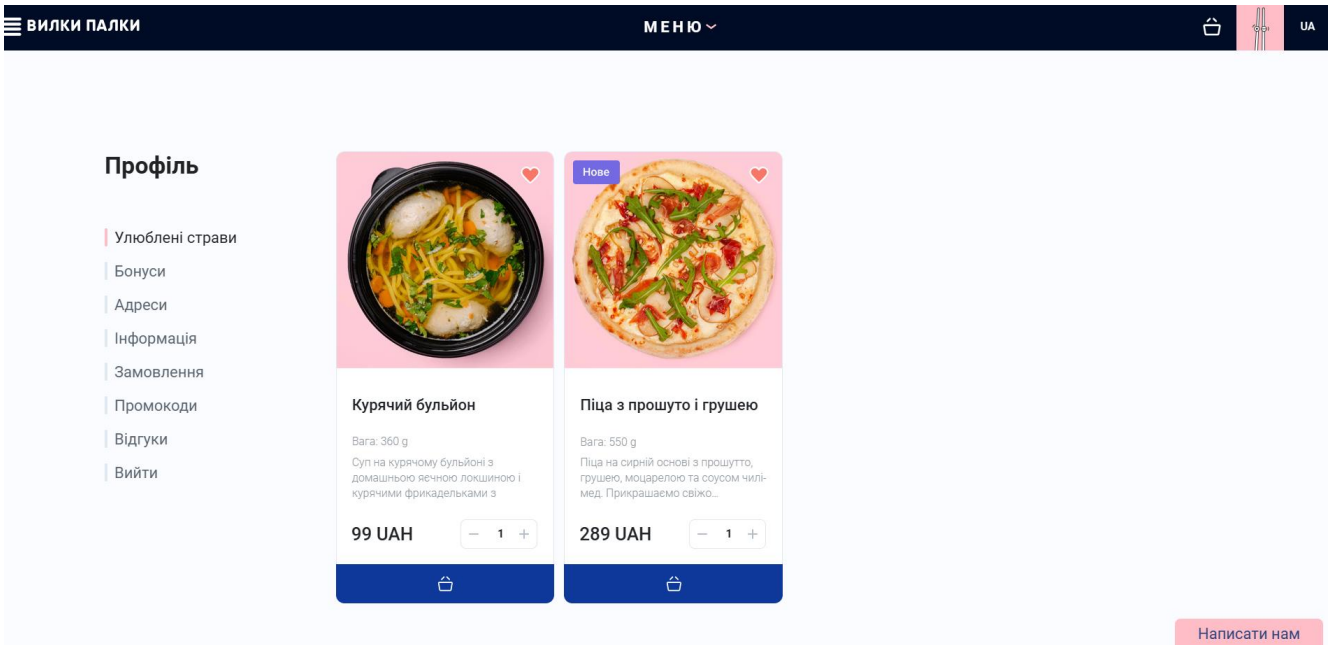


Рисунок 1.1 – Інтерфейс особистого кабінету: розділ «Улюблені страви»

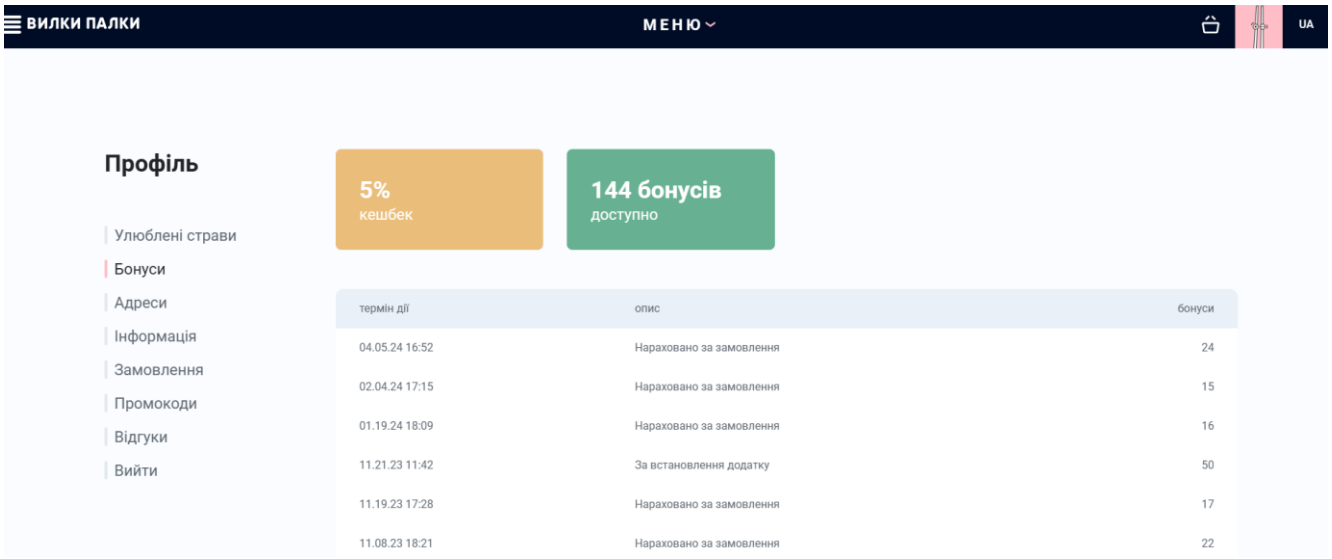


Рисунок 1.2 – Система бонусів в особистому кабінеті

Другий аналог – «SimpleFood». Цей застосунок для замовлення їжі, розроблений на PHP з використанням CMS WordPress та БД MySQL, також адаптивний під різні пристрої [3].

Щодо функціоналу, то він схожий з попереднім аналогом, також доступна реєстрація, історія замовлень, меню за розділами, та швидка реєстрація.

Але декілька головних *переваг*:

– можливість "збирати" блюдо (ланч), що дозволяє користувачам створювати свої власні комбінації страв;

– наявність мапи з цінами доставки.

Основні *недоліки*:

– незручно гортати меню на мобільній версії (у вигляді каруселі), що може залишити неприємний досвід у користувачів, як зображено на рисунку 1.3;

– відсутність фільтрів, що ускладнює процес пошуку конкретних страв або категорій;

– обмежена функціональність облікового запису, де доступні тільки налаштування та історія замовлень, може обмежувати можливості користувачів.

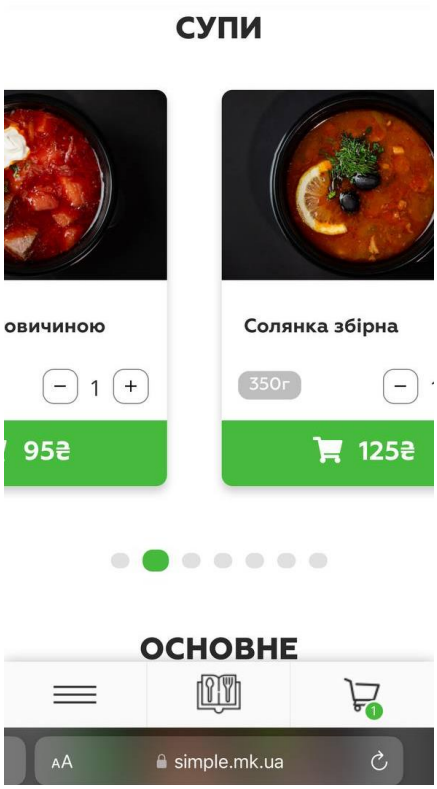


Рисунок 1.3 – Меню на мобільній версії

Також можна ще зазначити про незручність перегляду складу блюда, приклад зображено на рисунку 1.4;

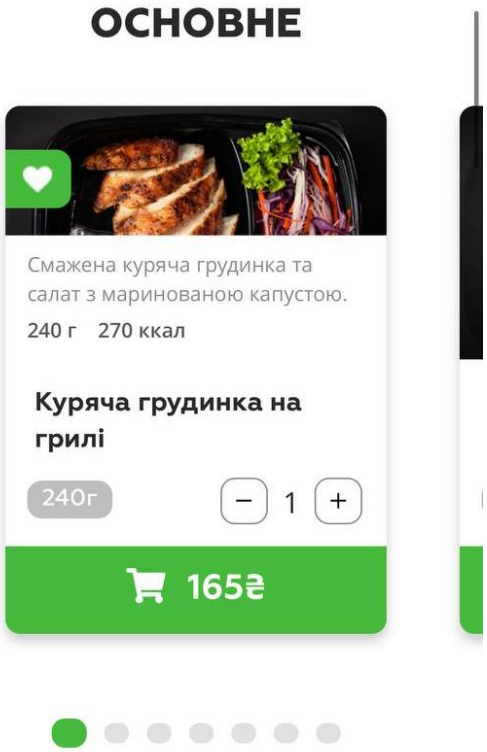


Рисунок 1.4 – Склад блюда

Третій аналог – онлайн-ресторан «NaDivanE» [4]. Особливістю та й недоліком є відсутність реєстрації. Загальний функціонал щодо оформлення та пошуку страв, та відображення меню – схожий, також є список обраних страв, після якого створюється замовлення. Але можна виділити певний ряд *недоліків*, а саме:

- відсутність відгуків;
- відсутність фільтрації та пошуку меню;
- не зручно обирати додаткові параметри до страви (соуси), як зображено на рисунку 1.5;
- відсутність фільтрів, що ускладнює процес пошуку конкретних страв або категорій;

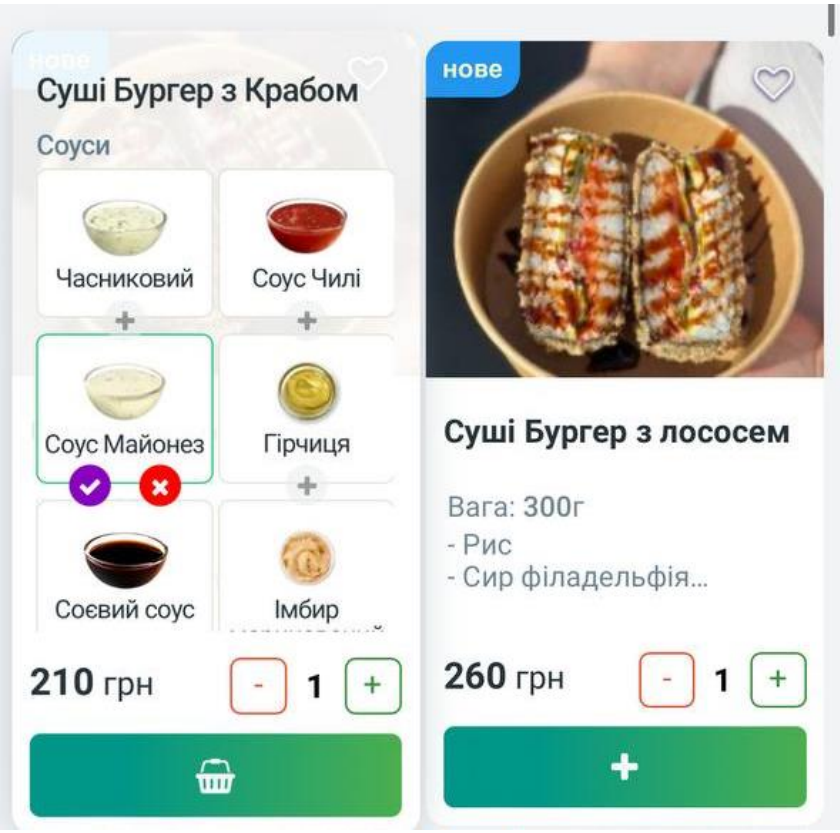


Рисунок 1.5 – Додаткові соуси до страви

1.3 Специфікація вимог

Розробка вебзастосунку для покращення ефективності роботи кафе навинос. Окрім розробки самого сайту з меню, особистим кабінетом, доставкою, це ще створення панелі адміністратора для редагування меню, контроль та керування замовлень, аналітика, система бонусів.

Клієнтам будуть доступі наступні функції:

- перегляд меню;
- фільтрація та пошук страв;
- реєстрація та авторизація;
- оформлення та оплата замовлення;
- накопичення та оплата бонусами;
- додавання та редагування складників в замовленні;
- застосування промокодів;
- додавання улюблених страв;
- написання відгуків.

Адміністратору буде надано можливості:

- редагування цін;
- редагування опису страв та напоїв;
- створення опцій (наприклад, соуси, додавання цукру в каву та інше);
- видалення відгуків.

Сценарії роботи системи:

1) Клієнт може відстежувати статус свого замовлення через особистий кабінет на вебсайті або застосунку.

2) Адміністратор може відстежувати популярність страв, змінювати налаштування для опцій страви, та змінювати статус замовлення.

3) Додавання товарів до кошика: користувач обирає страви та напої, що сподобалися з меню, і додає їх до кошика.

4) Оформлення замовлення.

5) Реєстрація на сайті.

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Призначенням ПЗ, що розробляється, є автоматизація замовлень, ведення обліку, та забезпечення користувачів доступом до меню, особистого кабінету та всього функціоналу для замовлення.

Погодження, що ухвалені в програмній документації

Погоджень не ухвалено.

Межі проєкту ПЗ

Крайня дата завершення роботи над ПЗ – 14.06.2024р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Дане ПЗ не має певних специфікацій, використовується як в повсякденному житті, так й на певні свята або для організацій.

Характеристики користувачів

Основні характеристики користувачів: наявність будь-якого девайсу (ПК, смартфон, планшет) та доступ до мережі Інтернет.

Загальна структура і склад системи

1) Головна сторінка: основні складові меню, розділ з популярними стравами та способи зв’язку.

2) Сторінка «Меню»: доступ до страв, наявність фільтрів, модальне вікно для зазначення додаткових параметрів при замовленні.

3) Сторінка «Про нас» – інформаційна.

4) Сторінка «Мій акаунт»: розподіляється на «Улюблене», «Персональні дані» та «Історія замовлень». На першій відображається перелік обраних страв та напоїв. На другій – особисті дані, з можливістю змінити, а на третій – історія замовлень, склад замовлення, загальна сума та скільки бонусів списано.

5) Сторінки авторизації та реєстрації в системі.

Загальні обмеження

Обмеження для роботи з ПЗ – наявність доступу до мережі Інтернет та наявність дійсного номеру телефону.

ФУНКЦІЇ ВЕБЗАСТОСУНКУ ПЛАНУВАННЯ ПОДОРОЖЕЙ

Функція замовлення страв

Опис функції

Функція дозволяє обрати в меню потрібні страви, а на сторінці кошику, вже оформити замовлення, ввести необхідні дані, та обрати спосіб доставки.

Вхідна і вихідна інформація

Вхідна інформація – список страв, ім’я, телефон, адреса, спосіб доставки та оплати, за наявністю введення промокоду та додатковий коментар.

Вихідна інформація – створене замовлення, переглянути можна в особистому кабінеті.

Функціональні вимоги

Доступ до мережі Інтернет.

Функція пошуку страв та застосування фільтрів

Опис функції

Функція пошуку допомагає швидше знайти бажану страву, а завдяки фільтрам можна відсортувати результати або задати біль детальні вимоги.

Вхідна і вихідна інформація

Вхідна інформація – назва / категорія / ціна / калорійність страви.

Вихідна інформація – страви, що задовольняють критерії пошуку.

Функціональні вимоги

БД з усіма блюдами, що в наявності.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

Для цього ПЗ джерелом вхідної інформації є користувач. Користувач має задавати необхідні дані для пошуку страви, а також вводить дані для входу та реєстрації. Для входу – це номер телефону та пароль.

Вимоги до способів організації, збереження та ведення інформації

Для цього вебзастосунку обрана БД MySQL.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Конкретних вимог до технічного забезпечення немає. Вебсайт адаптивний, тому правильність роботи забезпечується на усіх пристроях з сучасним браузером.

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Архітектура складається з клієнтської та серверної частин, та БД.

Системне програмне забезпечення

Для вебзастосунку обрано фреймворк Laravel, а стосовно БД – MySQL.

Програмне забезпечення ведення інформаційної бази

CRUD-операції виконуються через БД.

Мова і технологія розробки ПЗ

Розробка ПЗ з фреймворком Laravel. Мова розробки – PHP.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс користувача повинен бути зрозумілим, простим та згідно з правил в UI/UX дизайні, задля ефективності роботи.

Апаратний інтерфейс

Апаратним інтерфейсом є ПК або мобільний пристрій користувача для роботи з вебзастосунком.

Програмний інтерфейс

Laravel – популярний фреймворк для створення вебзастосунків на мові PHP, спрощує процес розробки за роботи з БД.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

ПЗ доступне та зручне для будь-якого користувача з девайсом та доступом до мережі Інтернет.

Супроводжуваність

Вебзастосунок не потребує супроводжуваності.

Переносимість

ПЗ має працювати як на ПК, так й на мобільних пристроях та планшетах.

Продуктивність

Ефективність роботи програмного забезпечення залежить від швидкості доступу до мережі Інтернет.

Висновки до розділу 1

В першому розділі КРБ розглянуто вебзастосунок для кафе навинос, визначені головні ролі системи, приклади сценаріїв роботи. Описано функціонал вебсайту, а в аналізі існуючих систем виявлено ключові аспекти та вдалі рішення, які були враховані в проєктуванні майбутньої системи, серед аналогів були онлайн-ресторани «Вилки-Палки», «NaDivanE», «SimpleFood».

2 ПРОЄКТУВАННЯ ВЕБСАЙТУ ДЛЯ КАФЕ НАВИНОС

2.1 Сценарії використання вебзастосунку

Діаграма варіантів використання є корисним інструментом для візуалізації функціональності системи. А сценарій – це послідовність дій, що й як користувач взаємодіє з системою задля досягнення певного результату. На діаграмі відображені різні сценарії використання системи, що допомагає зрозуміти, як користувачі будуть взаємодіяти з системою та які операції вони можуть виконати.

Коротка форма: додавання товарів до кошика: користувач вибирає страви та напої, що сподобалися, з меню і додає їх до кошика.

Клієнт відкриває сторінку меню на сайті кафе, де представлені доступні страви та напої з їх описом та цінами. Після вибору товару клієнт натискає кнопку «Додати в кошик» поряд із вибраною стравою або напоєм. Система додає вибраний товар у кошик. При додаванні кожного товару у кошик система автоматично розраховує загальну вартість замовлення на основі цін доданих товарів. Після додавання товарів до кошика клієнт може повернутися до перегляду меню та додати ще товари. Клієнт завершує процес покупок, коли додані усі потрібні товари в кошик, і натискає кнопку «Оформити замовлення».

Поверхнева форма: оформлення замовлення.

Головний сценарій (успішний):

Користувач переглядає меню та вибирає бажані страви та напої. Після вибору, він переходить до корзини, де переглядає свій вибір та може внести зміни. Після цього він натискає кнопку «Оформити замовлення». Система перевіряє наявність обраних товарів. Потім користувач вводить необхідні дані для доставки або самовивозу: адресу, контактні дані тощо. Після підтвердження замовлення система приймає платіжну інформацію, якщо це потрібно. Після успішної операції користувач отримує підтвердження замовлення та відображається його номер замовлення.

Альтернативні сценарії:

1) Використання промокоду

Користувач вводить промокод для отримання знижки. Система перевіряє дійсність промокоду. Промокод використовується повторно – виводиться помилка та промокод не буде застосований.

2) Втрата зв'язку під час оформлення замовлення

Користувач розпочинає процес оформлення замовлення на сайті, але в середині процесу втрачає зв'язок з інтернетом або сервером сайту виходить з ладу, що призводить до втрати даних і невдалого оформлення замовлення.

Повна форма

Таблиця 2.1 – Повний сценарій

Use Case Name	Реєстрація на сайті
Scope	Система кафе навинос для прийому замовлень
Level	User-goal
Primary Actor	Користувач
Stakeholders and interests	– Користувач: бажає створити обліковий запис на сайті для замовлення страв та отримання знижок або пропозицій. – Адміністратор сайту: зацікавлений у відповідному функціонуванні системи реєстрації користувачів та забезпеченні їхньої безпеки. – Власник кафе: зацікавлений в правильній роботі системи, та залучення клієнтів.
Preconditions	– Система працює коректно. – Користувач має доступ до Інтернету.
Main Success Scenario	1) Користувач відвідує сайт. 2) Користувач натискає кнопку "Зареєструватися" на головній сторінці сайту. 3) Система відображає форму реєстрації з полями для введення ім'я, електронної адреси, пароля тощо. 4) Користувач вводить свої особисті дані в поля форми реєстрації.

Кінець таблиці 2.1

Main Success Scenario	5) Користувач підтверджує свою реєстрацію, натискаючи кнопку «Зареєструватися». 6) Система перевіряє введені дані на відповідність формату та наявність унікальної електронної адреси. 7) Якщо дані введені коректно і адреса електронної пошти є унікальною, система створює обліковий запис користувача. 8) Після цього користувач може використовувати свій обліковий запис для входу. 9) Користувач теперь може використовувати увесь функціонал.
Extensions	– Користувач вводить неправильну або неповну інформацію під час реєстрації. – Користувач намагається створити обліковий запис з електронною адресою, яка вже існує в системі. – Система стикається з технічною проблемою під час створення облікового запису користувача.
Miscellaneous (Open Issues)	Реалізація функціонала для користувачів з можливістю управління своїм обліковим записом, включаючи зміну особистих даних, пароля, налаштувань тощо.

Діаграма варіантів використання вебзастосунку представлена на рисунку 3.1.

2.2 Розробка діаграми послідовностей

Діаграма послідовності (Sequence Diagram) використовується для моделювання логіки сценаріїв використання, демонструючи передачу інформації між об'єктами в системі під час виконання сценарію. Вона показує цю інформацію в горизонтальному та вертикальному вирівнюванні, а об'єкти, що надсилають запити, відображені у вигляді блоків [5].

Основними компонентами моделей бізнес-об'єктів є зовнішні і внутрішні виконавці, а також бізнес-суті, відображають все, що використовують внутрішні

виконавці для реалізації бізнес-процесів. На діаграмі послідовностей для деякого набору об'єктів на єдиній тимчасової осі показані життєвий цикл і взаємодія (відправка запитів і отримання відповідей) [6]. Основними елементами діаграми є позначення об'єктів, вертикальні часові лінії, діяльності об'єкта або виконання певної функції, і стрілки обміну сигналами або повідомленнями між об'єктами. Результат діаграми відображено на рисунку 2.1.

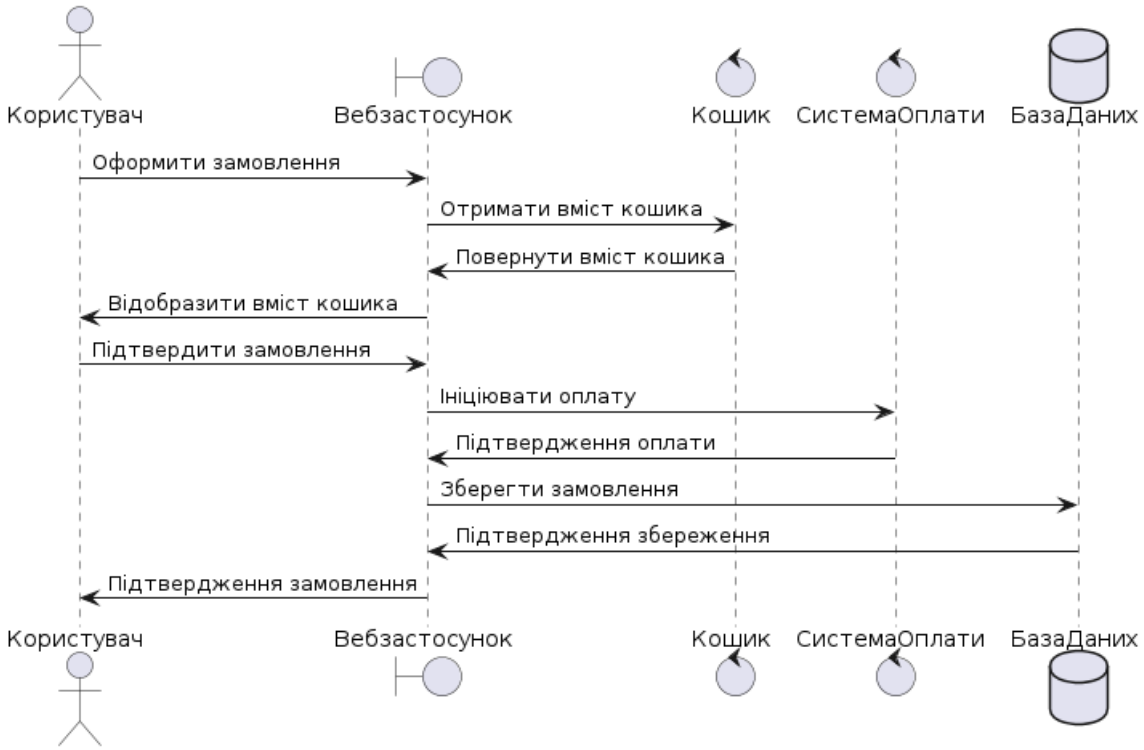


Рисунок 2.1 – Діаграма послідовності оформлення замовлення

Процес оформлення замовлення відображено на діаграмі послідовності: тобто спочатку створюється запит щодо вмісту кошика, потім йде ініціація користувачем оплати на цю суму, та система оплати повертає підтвердження. В кінцевому результаті з БД повертається підтвердження та користувач отримує на інтерфейсі підтвердження.

Взаємодія з адмін-панелю, а саме редагування опису страви відображено на рисунку 2.2.

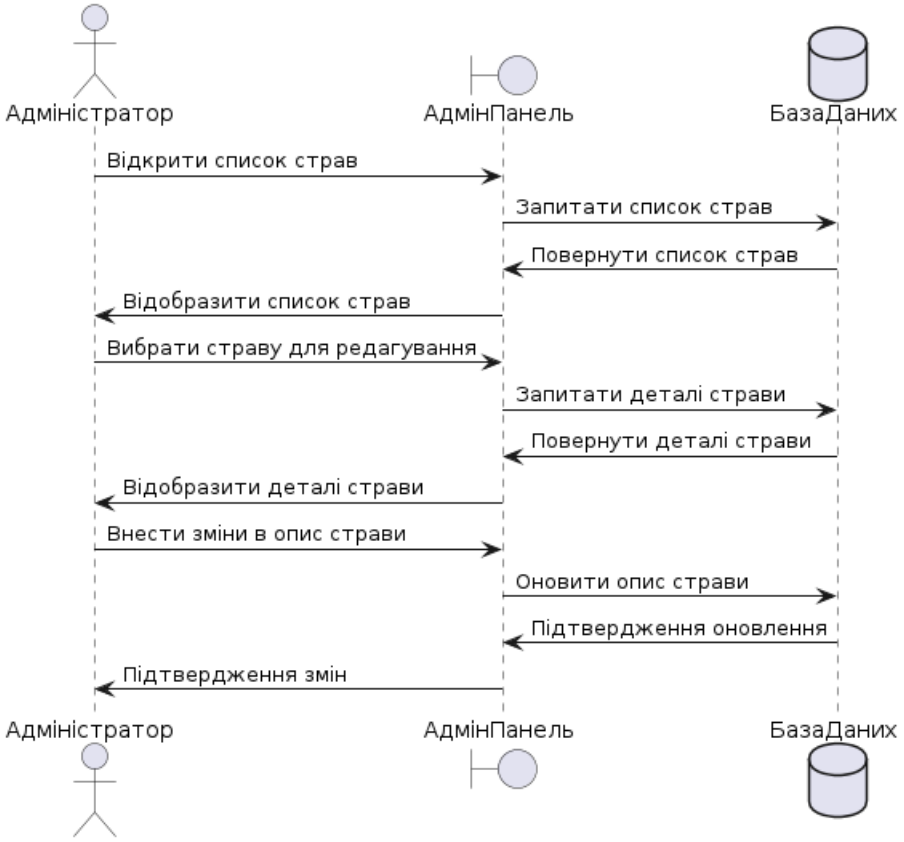


Рисунок 2.2 – Діаграма редагування страви в меню

Після авторизації адміністратор обирає необхідну страву та до БД відправляється запит щодо детальної інформації обраної позиції в меню. Потім формується запит на оновлення опису або іншого розділу, і відправляється знову на сервер. В БД оновлюються дані та в відповіді – підтвердження успішного оновлення. Ця діаграма чтко показує, як відбувається процес редагування опису від вибору до успішного результату.

2.3 Розробка мокапів системи

Мокап – це деталізований ескіз інтерфейсу користувача (UI) програми, який допомагає узгодити всі важливі аспекти [7]. Мокапи дозволяють візуалізувати концепцію та структуру майбутнього сайту, що допомагає краще зрозуміти, як виглядатиме кінцевий продукт. Попередня обробка дизайну та функціоналу у вигляді мокапів допомагає знизити ризики непорозуміння вимог та очікувань клієнта.

При розробці мокапа важливо враховувати кілька ключових аспектів, щоб створити ефективний дизайн, який буде основою до розробки сайту. Основні моменти, які слід враховувати:

– цілі та завдання проєкту;
– логічну структуру сторінок і навігацію;
– візуальні елементи та виділення важливої інформації, єдиний стиль;
– обмеження введення даних.

Під час проєктування вебсайту доставки їжі розроблено наступні мокапи сторінок:

– головної сторінки застосунку представлено на рисунку 2.5;
– сторінки каталогу представлено на рисунку 2.6;
– сторінки каталогу оформлення замовлення представлено на рисунку 2.4;
– сторінки особистого кабінету користувача зображено на рисунку 2.7;
– сторінки меню, а саме застосування фільтрів при пошуку зображено на рисунку 2.9.

Відображенням верхнього головного меню та прикладу заповненого кошику є рисунок 2.3.

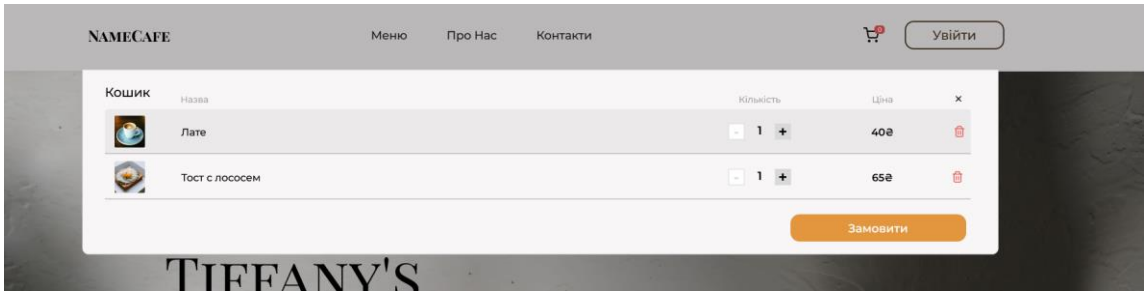


Рисунок 2.3

Верхнє меню містить основні навігаційні елементи, такі як меню, про нас та контакти, також є відразу доступ до акаунту або просто кнопка для реєстрації. У кошику можна відразу змінити кількість позицій, видалити позицію або відразу очистити усе.

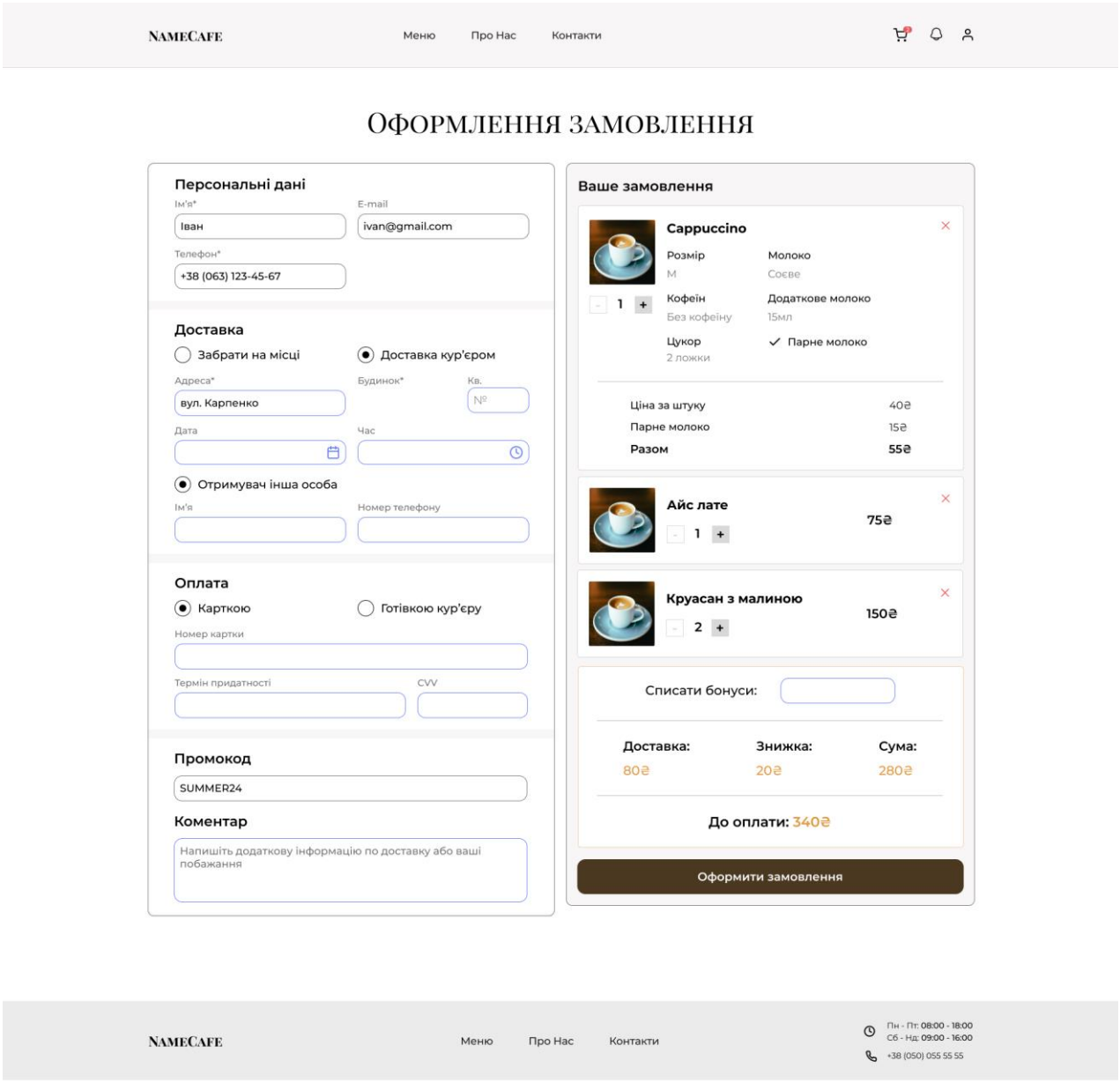


Рисунок 2.4 – Мокап сторінки оформлення замовлення

Як раніше зазначено, на головній сторінці є відразу розділ з популярними стравами на основі кількості вподобайок авторизованих користувачів, розділ відгуків з різних позицій та розділ для зв’язку, також до нього є доступ з хедеру сайту.

Сторінка оформлення замовлення візуально поділена на 2 частини задля структуризації даних: ліворуч маємо поля для заповнення, праворуч – замовлення користувача, додаткову інформацію якого можна дізнатися після натискання.

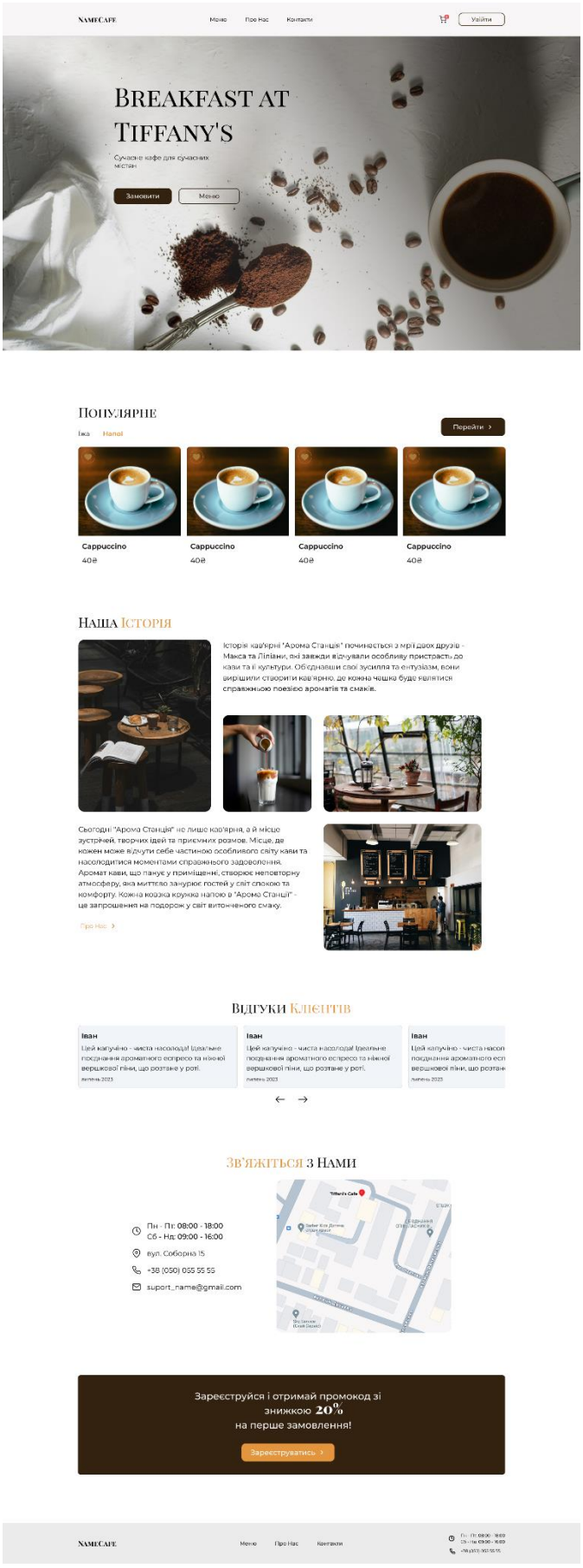


Рисунок 2.5 – Мокап головної сторінки застосунку

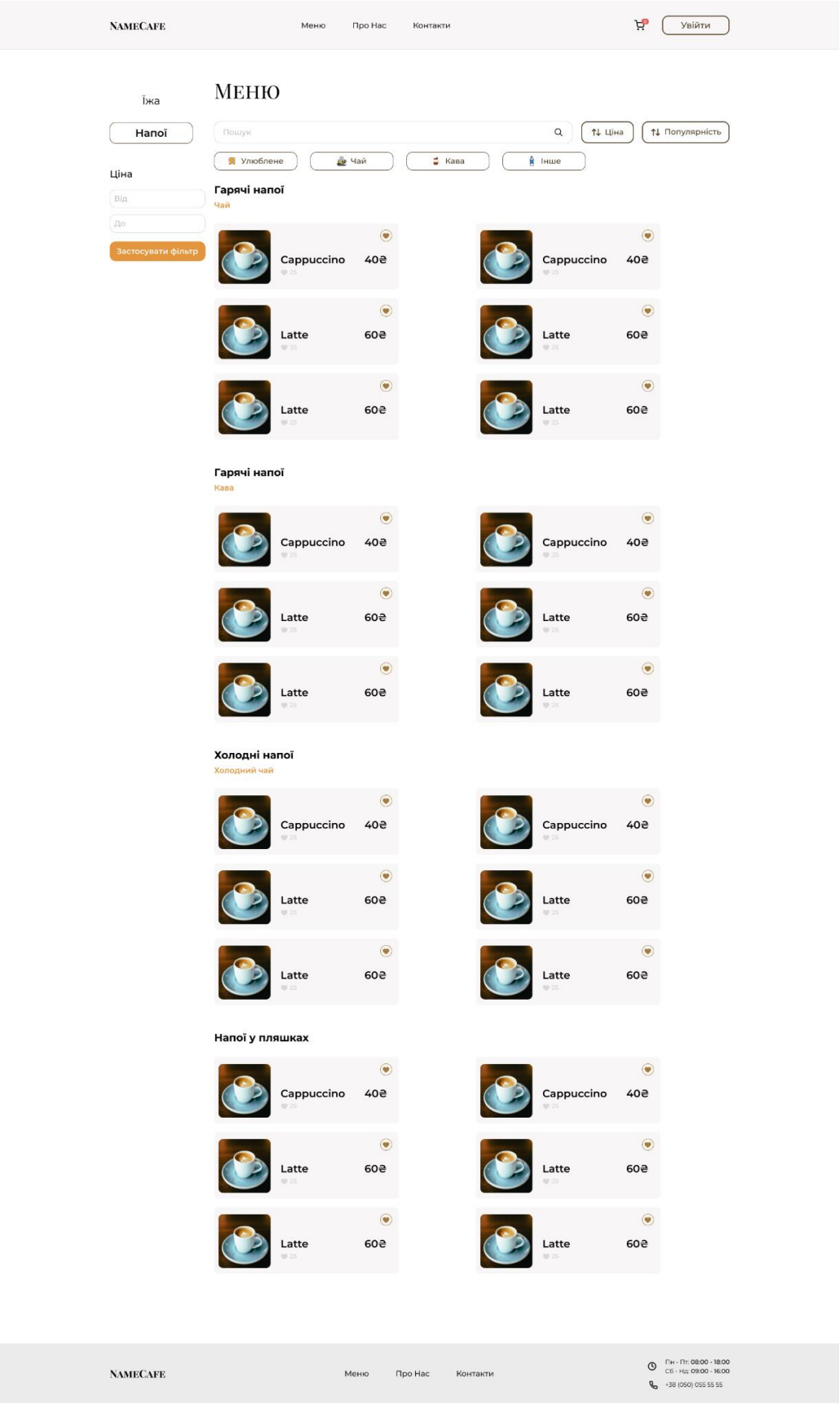


Рисунок 2.6 – Мокап сторінки каталогу

Також можна одразу в замовленні застосувати промокод та списати бонуси. Оплата карткою представлена, а стосовно готівки – там поля для введення суми, з якої потрібна решта для кур’єра.

Особистий кабінет має мінімальне меню з функціоналом: передивитись улюблені страви та напої, історію замовлень (приклад на рисунку 2.8), де видно скільки отримано та списано бонусів, а при натисканні буде уся інформація.

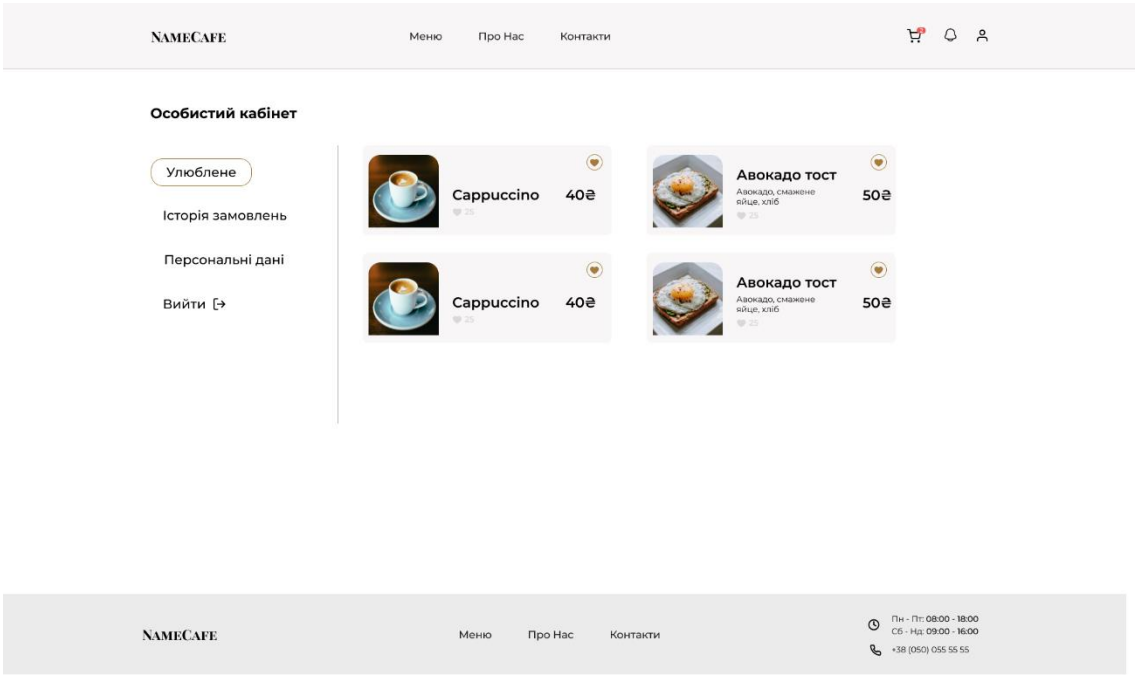


Рисунок 2.7– Мокап особистого кабінету користувача

А в розділі «Персональні дані» відображено поля з ім’ям, телефоном, поштою, адресою та можливістю змінити пароль.

Рисунок 2.8 – Відображення бонусів та коротко інформації про виконане замовлення

Фільтри для замовлення є в боковій панелі: є можливість задавати ціни та калорійність страви, а зверху є поле для пошуку та фільтрація за підрозділами.

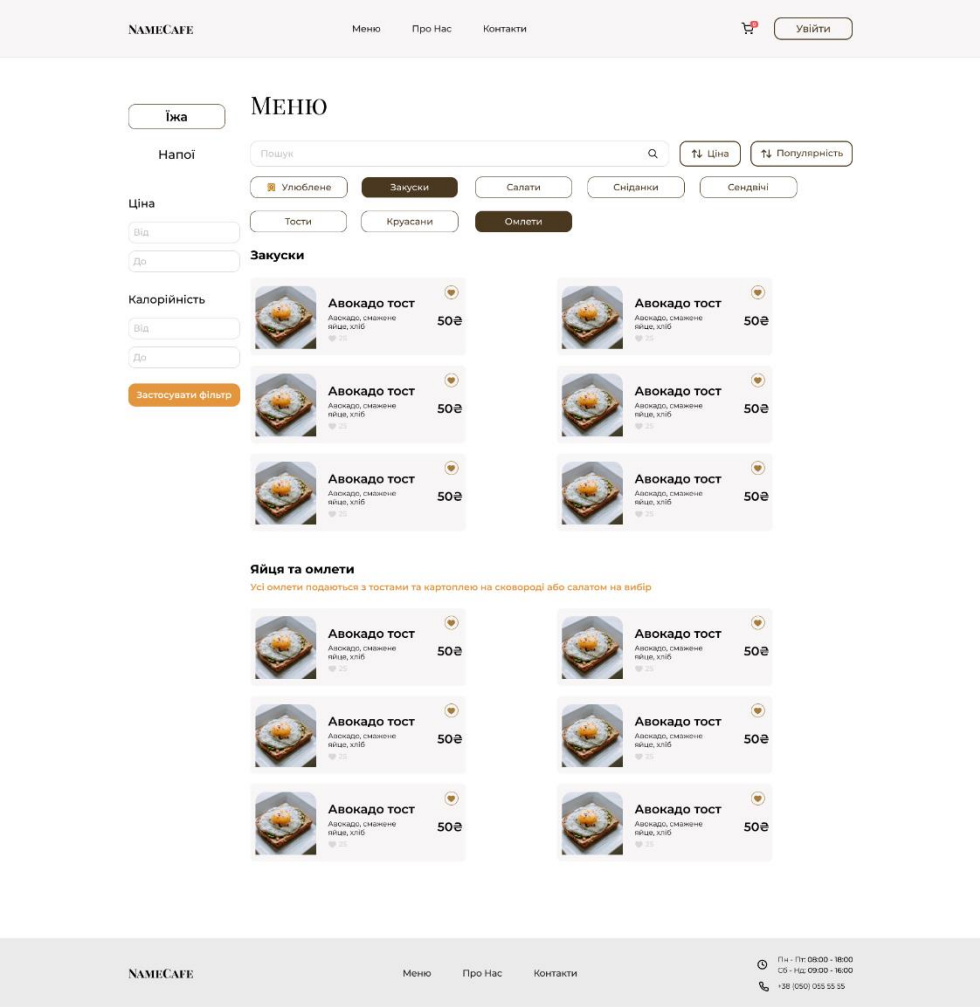


Рисунок 2.9 – Мокап сторінки застосування фільтрів в каталозі

Також важливо зазначити, що при натисканні на страву з’являється модальне вікно з карточкою товару, де одразу можна переглянути склад, калорійність, відгуки та додати до кошику.

2.4 Розробка моделі потоків даних системи

Діаграма розгортання – це діаграма, яка показує розташування компонентів програмної системи та середовища, в якому вони працюють. Вона відображає архітектурну структуру системи та зв'язки між її компонентами на рівні обладнання. Діаграму розгортання системи відображено на рисунку 2.10.

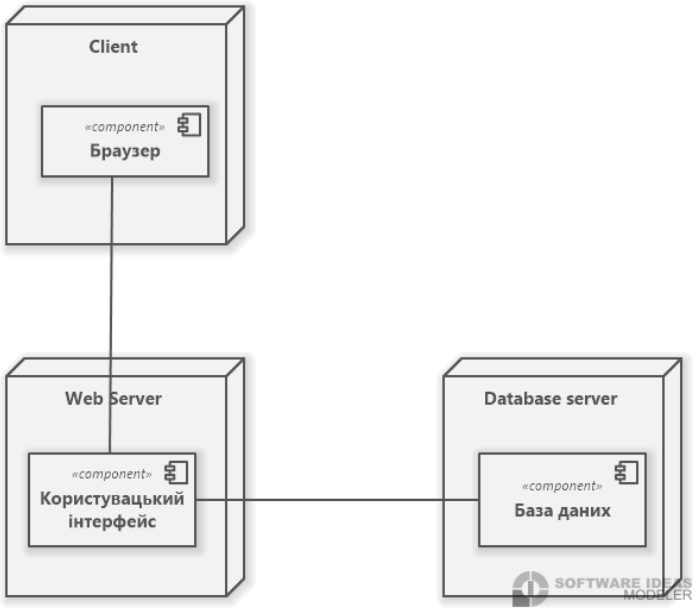


Рисунок 2.10 – Діаграма розгортання

Клієнт через ПЗ, встановлене на пристрої має доступ до вебсайту кафе. Браузер надсилає запити на сервер та відображає отримані дані та відповіді. Web Server на якому розміщені файли сайту для відображення інтерфейсу, приймає та обробляє запити з браузера. Database server де зберігаються всі необхідні дані про меню, замовлення, користувачів та інші для роботи сайту.

2.5 Розробка діаграми станів системи

Діаграма станів – це діаграма, яка моделює станові переходи об'єктів в системі в залежності від стану об'єкта та зовнішніх подій. Вона використовується для опису поведінки об'єкта та відображення всіх можливих станів, у які може переходити об'єкт під час виконання програми.

Діаграми станів для системи зображені на рисунках 2.11-2.12.

Перша діаграма стосується навігації на сайті: початковим станом є головна сторінка, після чого є варіанти, наприклад, це перехід до меню, потім до сторінки кошику й далі оформлення замовлення. Або ж просто перегляд своїх даних в кабінеті.

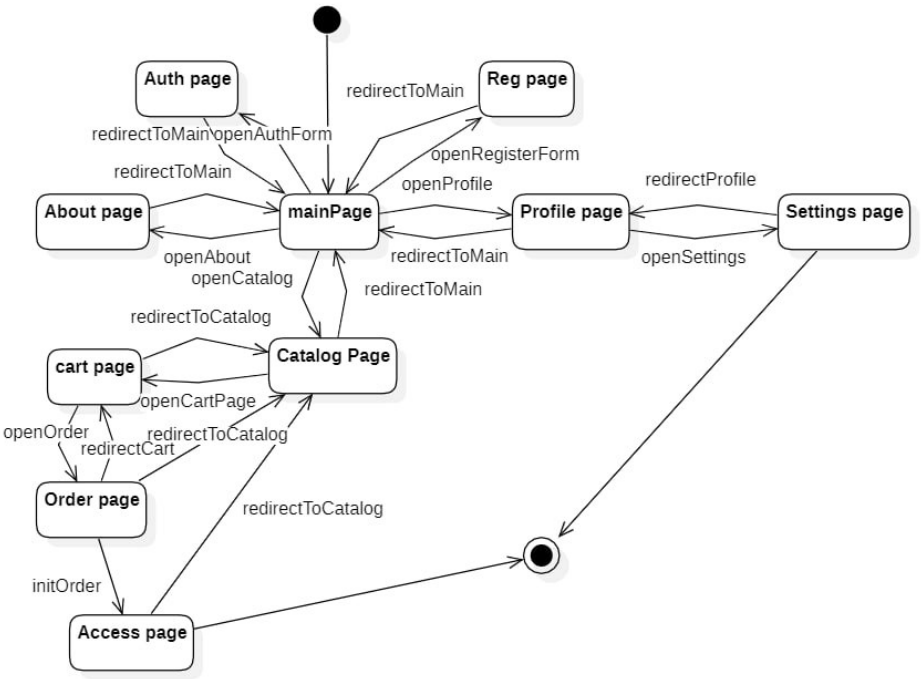


Рисунок 2.11 – Діаграма станів системи

Друга діаграма стосується вже оновлення даних: де спочатку доступ до редагування, а потім перехід на підтвердження змін та їх успішне оновлення.

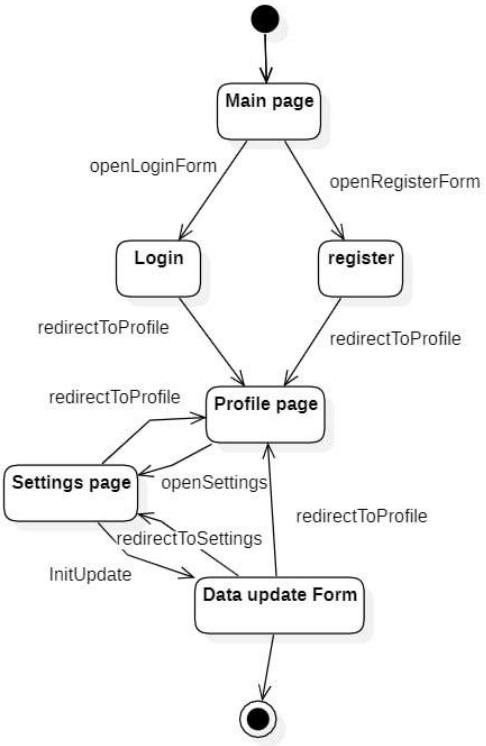


Рисунок 2.12 – Оновлення даних о користувачеві

Завдяки діаграмі станів можна виявити вразливі місця і потенційні проблеми в процесі обслуговування, що дозволяє оптимізувати операції і покращити загальну роботу кафе. Також може бути частиною документації, яка описує поведінку системи, особливо це корисно для подальшого обслуговування та розвитку системи. Також завдяки цьому можна врахувати обробку виняткових ситуацій, таких як скасування замовлення, повернення коштів, що допомагає забезпечити стійкість системи.

Висновки до розділу 2

У другому розділі КРБ розглянуто процес проєктування вебзастосунку для кафе навинос. Проєктування включало розробку різноманітних моделей та діаграм, які описують функціональні та архітектурні вимоги системи. Розроблено діаграму варіантів використання системи, оскільки надає уявлення про сценарії використання застосунку та ролі користувачів, які можуть взаємодіяти зі системою. Діаграма класів системи представляє основні класи та їх взаємозв'язки. Модель розгортання системи описує фізичну архітектуру системи. Загалом аналіз цих аспектів дозволяє більш детально продумати функціонал та усі можливі сценарії застосунку.

3 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис технологій розробки

MySQL, PHP (зокрема Laravel), TailwindCSS і PhpStorm є ключовими інструментами сучасного веброзробника. Вони разом створюють потужний та інтегрований стек технологій, що дозволяє створювати складні, функціональні та ефективні вебзастосунки. Кожен з цих інструментів має свої унікальні особливості та переваги, які роблять їх незамінними у розробці.

MySQL є однією з найпопулярніших систем керування базами даних з відкритим кодом. Вона використовується для зберігання та керування даними в різних вебзастосунках. MySQL забезпечує високий рівень надійності та продуктивності, це є причиною підвищеного попиту серед багатьох розробників та компаній. Однією з ключових особливостей є її сумісність з різними операційними системами та інтеграція з популярними мовами програмування, такими як PHP, Java, Python та інші. Значною перевагою MySQL є її безпека. Це один із найважливіших критеріїв, на який потрібно звертати увагу, завдяки системі доступу можна забезпечити високий рівень безпеки. Також можливе використання зашифрованого пароля. Основна причина популярності MySQL – це підтримка мови SQL. Це означає, що будь-який запит повинен мати певну структуру, інакше база даних не зможе правильно його обробити. Основними завданнями є організація одночасного доступу до даних великої кількості користувачів, а також маніпуляція інформацією [8]. Запити вводяться у спеціальному терміналі, призначеному керувати БД. У MySQL, як і у більшості реляційних баз даних, зв'язки між таблицями відіграють ключову роль організації даних і забезпеченні їх цілісності. Зв'язки між таблицями показують, як дані можуть впливати інші. Основі типи:

– один до багатьох (One-to-Many) – це найпоширеніший вид зв'язку. Тобто один запис в одній таблиці може відповідати багатьом записам в іншій таблиці;

– один до одного (One-to-One) – простий вид зв'язку означає, що одному запису з даної таблиці відповідає лише один запис з іншої таблиці;

– багато до багатьох (Many-to-Many) – цей вид зв'язку потребує наявності проміжної таблиці для забезпечення зв'язку. У цьому випадку один запис у першій таблиці може відповідати багатьом записам у другій таблиці та навпаки.

Завдяки своїй масштабованості, MySQL підходить для різних за масштабом проєктів: від звичайних сайтів до великих корпоративних систем. Її структура на основі клієнт-серверної архітектури дозволяє ефективно працювати з великими обсягами даних, забезпечуючи швидкість та стабільність.

PHP, а точніше його фреймворк Laravel, є потужним інструментом для створення вебзастосунків, та надає розробникам зручні засоби для реалізації різних функціональних можливостей вебсайтів та застосунків. Однією з головних переваг Laravel є його зручний та зрозумілий синтаксис, що сприяє підвищенню продуктивності розробників [9]. Фреймворк підтримує такі сучасні концепції програмування, як MVC, що забезпечує розділення інтерфейсу користувача та бізнес-логіки для розробки веб-застосунків, задля виконання основних методів та чіткої реалізації структури [10]. Це робить код більш організованим та легшим у підтримці. Крім того, Laravel надає безліч вбудованих інструментів та пакетів, таких як Eloquent ORM для роботи з БД, система міграцій для керування схемою баз даних, а також потужні засоби для автентифікації, авторизації та валідації даних. Eloquent – це реалізація Active Record для операції, тобто техніка, яка обертає базу даних в об'єкти [11]. Все це значно спрощує та пришвидшує процес розробки.

Зараз доступно 113 бібліотек, що містять понад 1000 функцій. З PHP можна взаємодіяти з базами даних, маніпулювати інформацією та динамічно створювати сторінки, також із важливих можливостей є:

– керування Macromedia, зображеннями та форматами документів;

– автентифікація користувачів за обліковими даними;

– оцінка пароля на рівень вгадуваності, порівнявши його з мовними словниками та патернами [12];

TailwindCSS є утилітарним CSS-фреймворком, який дозволяє створювати стильні інтерфейси користувача без написання власних стилів з нуля. На відміну від традиційних фреймворків, він пропонує набір класів, що можна комбінувати для створення бажаного дизайну. Це дозволяє уникнути створення великих файлів стилів та сприяє зменшенню кількості перевизначень та конфліктів стилів. Однією з головних переваг TailwindCSS є його гнучкість та можливість налаштування під конкретні потреби проєкту. Розробники можуть легко змінювати кольори, відступи, шрифти та інші властивості, використовуючи конфігураційний файл. Використовуючи попередньо розроблені службові класи, можна забезпечити узгодженість всього сайту, знижуючи ризик помилок та невідповідностей [13]. TailwindCSS також підтримує принципи “mobile-first” дизайну, що забезпечує створення адаптивних інтерфейсів, які виглядають чудово на будь-яких пристроях.

PhpStorm є потужним інтегрованим середовищем розробки (IDE) для PHP та веброзробки. PhpStorm пропонує розробникам широкий набір інструментів та функцій, які допомагають підвищити продуктивність та зручність роботи. Однією з ключових особливостей PhpStorm є його кодовий редактор, який забезпечує підсвічування синтаксису, автозаповнення коду та підтримку різних мов програмування. Це дозволяє розробникам швидко визначати та виправляти помилки, а також пришвидшує процес написання коду. PhpStorm також інтегрується з різними системами контролю версій, такими як Git, що забезпечує зручне керування версіями проєкту та спільну роботу в команді. Крім того, надає потужні інструменти для налагодження та профілювання коду, що допомагає виявляти та виправляти проблеми продуктивності та логіки застосунку.

MySQL, PHP (Laravel), TailwindCSS та PhpStorm разом утворюють потужний стек технологій, що дозволяє розробникам розробляти високоякісні вебзастосунки. MySQL забезпечує надійне зберігання та керування даними, Laravel надає зручний та продуктивний фреймворк для розробки, TailwindCSS

спрощує процес створення стильних інтерфейсів, а PhpStorm забезпечує потужне інтегроване середовище для написання та налагодження коду. Використання цього стеку дозволяє створювати проєкти будь-якої складності, зберігаючи при цьому високу продуктивність та зручність розробки.

Таким чином ці інструменти – невід'ємні компонентами сучасної веброзробки. Вони забезпечують всіма необхідними інструментами для створення функціональних, надійних, адаптивних вебзастосунків та user-friendly інтерфейсу. Використовуючи цей стек технологій, розробники можуть швидко та ефективно реалізовувати свої ідеї, забезпечуючи високу якість та зручність.

3.2 Розробка діаграми класів

Діаграма класів – це структурна діаграма, яка відображає класи системи, їх атрибути та взаємозв’язки між ними [14]. По суті, моделювання структури програми та визначення взаємодії класів між собою. Діаграма класів для вебзастосунку доставки їжі представлено на рисунку 3.2.

На діаграмі представлено 14 класів, взаємопов’язаних між собою, головними сутностями серед яких є Order, User та Positions. В класі OrderHasPositionHasAddons зазначено особливі дані про замовлення, й він вже зв’язаний з класами PositionsHasOptions та OrderHasPositions, тобто які позиції в ньому та які додаткові дані є до страви або напою. Клас PositionsHasOptions зазначає інформацію про категорію «Їжа» в меню. Клас Order відображає основну інформацію про замовника, час, тип оплати та доставки, також перевірка на застосування промокоду. Клас User має стандартні дані необхідні для авторизації на сайті та створення замовлення. Також важливо зазначити, що усі позиції в меню мають поля «likes» та «review», адже саме за кількістю лайків формується розділ на головній сторінці про популярні страви. Таким чином, діаграма класів – це остаточний результат проєктування і відправна точка процесу розробки [15].

3.3 Моделювання бази даних

База даних кафе – це система багатофункціональна та зручна. Тому на етапі моделювання бази даних, згідно з поставленим завданням, були виділено основні сутності, об'єкти з яких буде складатися дані в таблицях. Основною метою проєктування БД є скорочення надмірності збережених даних, а отже, економія об'єму використовуваної пам'яті, зменшення витрат на багаторазові операції.

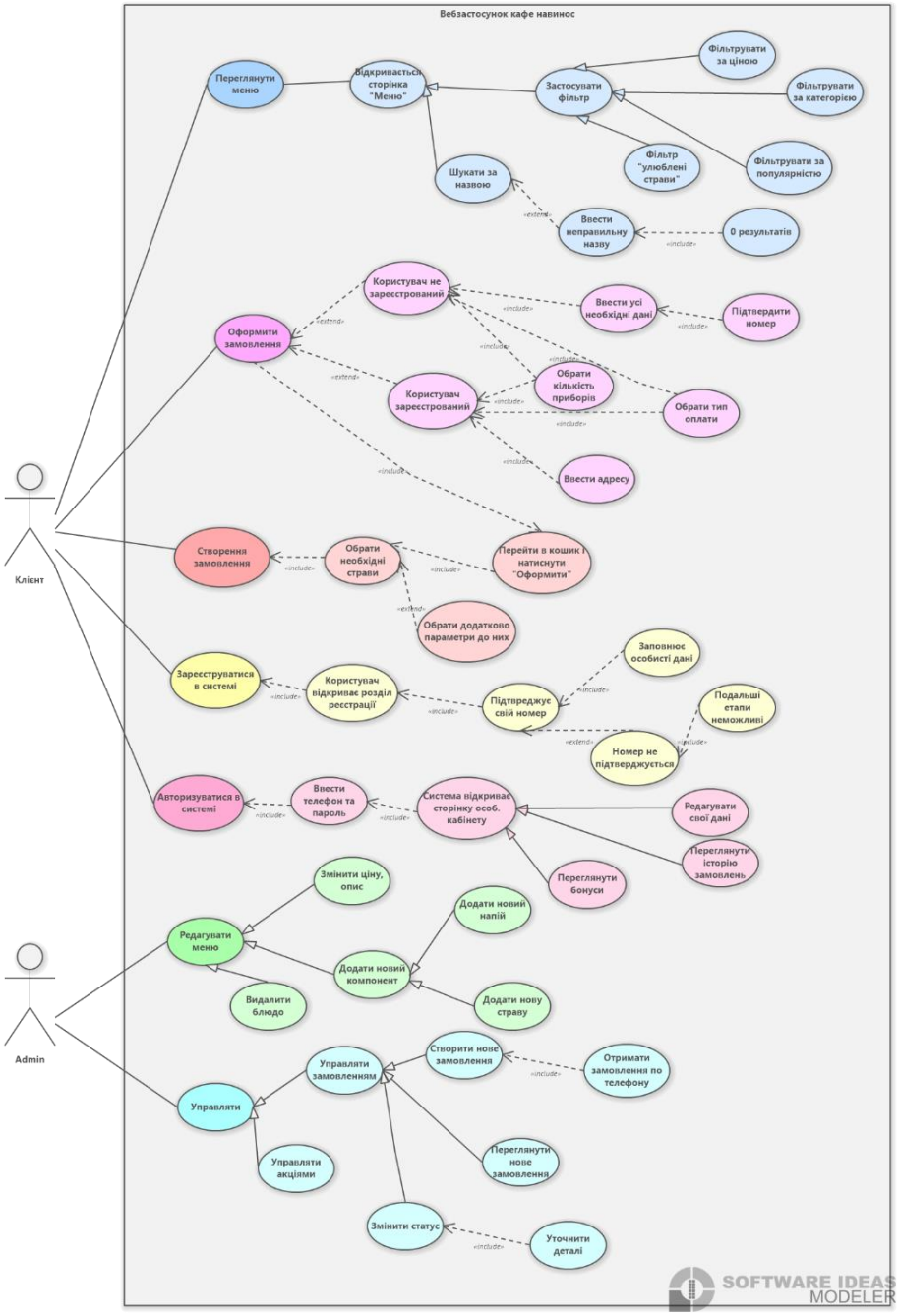


Рисунок 3.1 – Діаграма варіантів використання вебзастосунку

Результатом проєктування бази даних є перетворення опису й вимог предметної області у внутрішню схему БД [16].

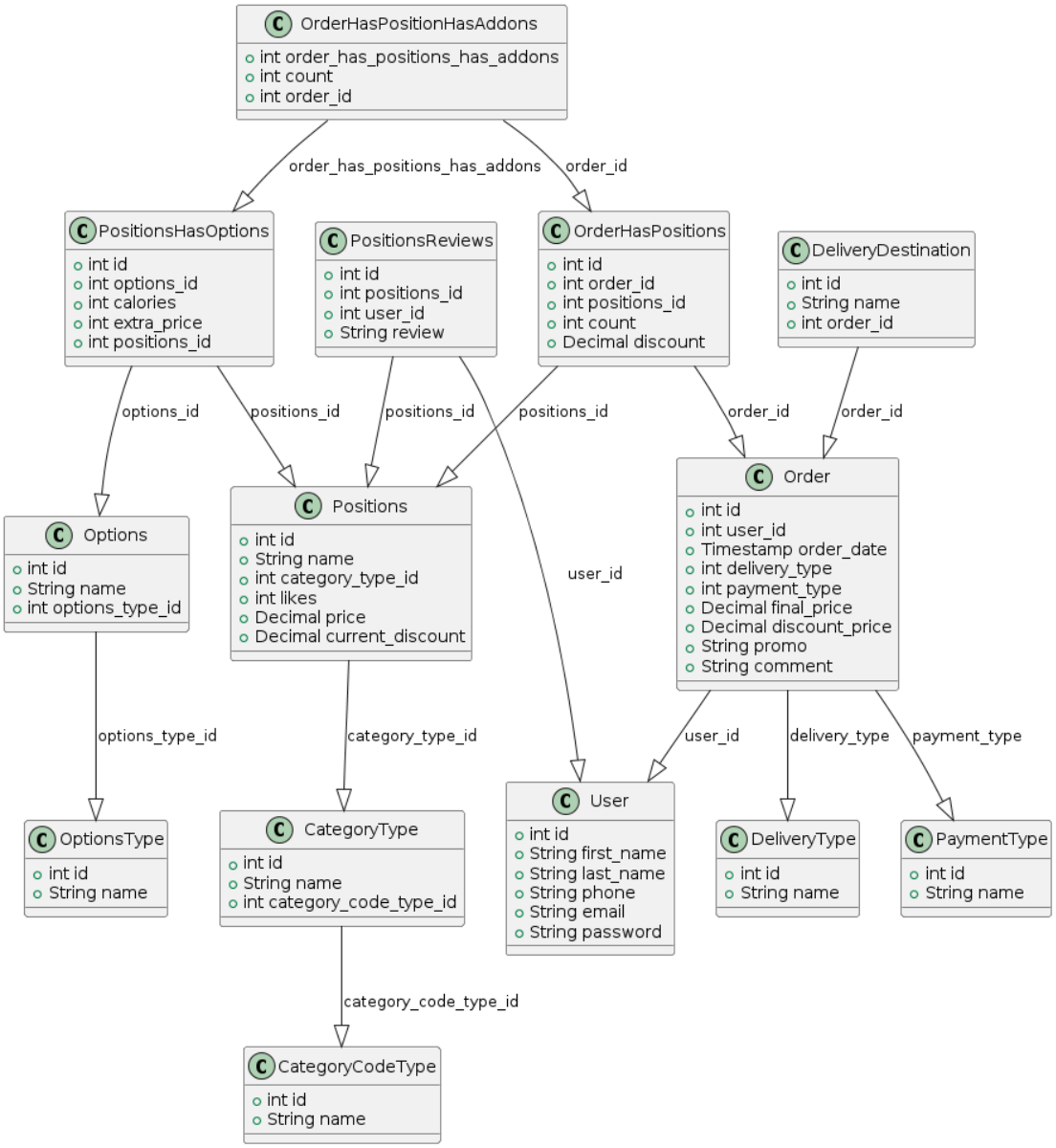


Рисунок 3.2 – Діаграма класів вебзастосунку

Фізична модель БД є конкретним втіленням логічної моделі даних в організації даних. Вона визначає, як дані фізично зберігаються та доступні на пристроях зберігання інформації, враховуючи різні технологічні аспекти, такі як файлові структури, індекси та інші параметри.

Модель БД кафе навинос представлено на рисунку 3.3.

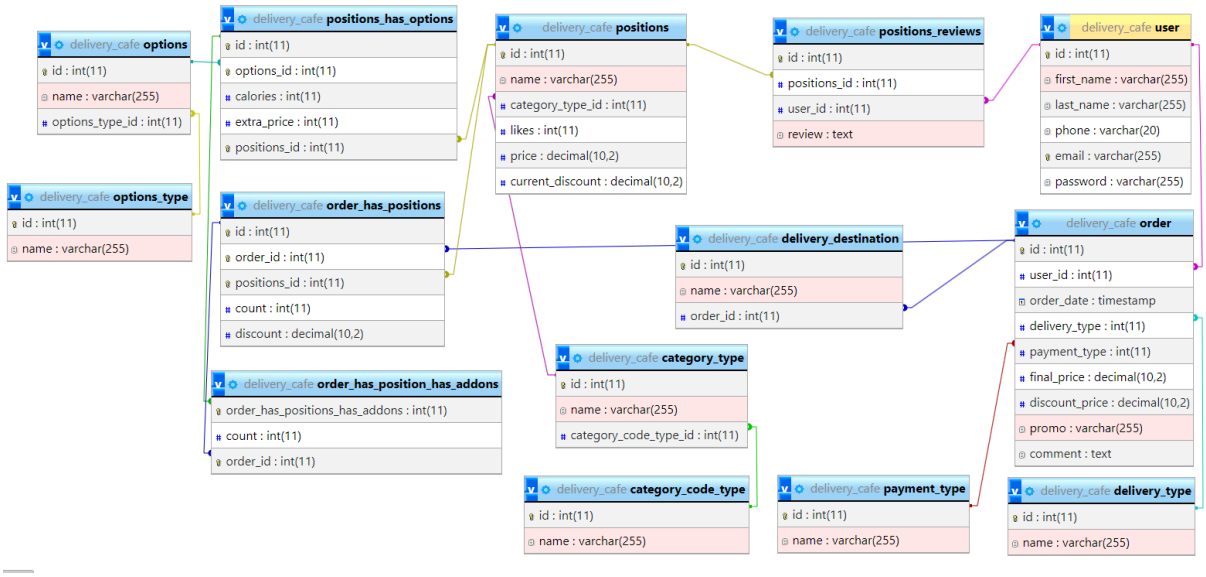


Рисунок 3.3 – Фізична модель бази даних

Тепер необхідно розглянути кожну таблицю:

Таблиця **Options** містить параметри для різних пунктів картки товару чи певних опцій, які можуть бути додані до позицій. Кожна опція має унікальний ідентифікатор (id), назву (name) та зв'язок з типом опцій (options_type_id). Структура таблиці представлена на рисунку 3.4.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	*Немає*		AUTO_INCREMENT	Більше
2	name	varchar(255)	*utf8mb4_unicode_ci*		Ні	*Немає*			Більше
3	options_type_id	int(11)			Ні	*Немає*			Більше

Позначити все *Вибрані:*

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

Додати 1 стовпець(ів) після options_type_id Виконати

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	options_type_id	BTREE	Ні	Ні	options_type_id	0	A	Ні	

Рисунок 3.4 – Структура таблиці Options

Таблиця **OptionsType** зберігає типи опцій. Кожен тип опцій має унікальний ідентифікатор (id) та назву (name). Структура таблиці представлена на рисунку 3.5.

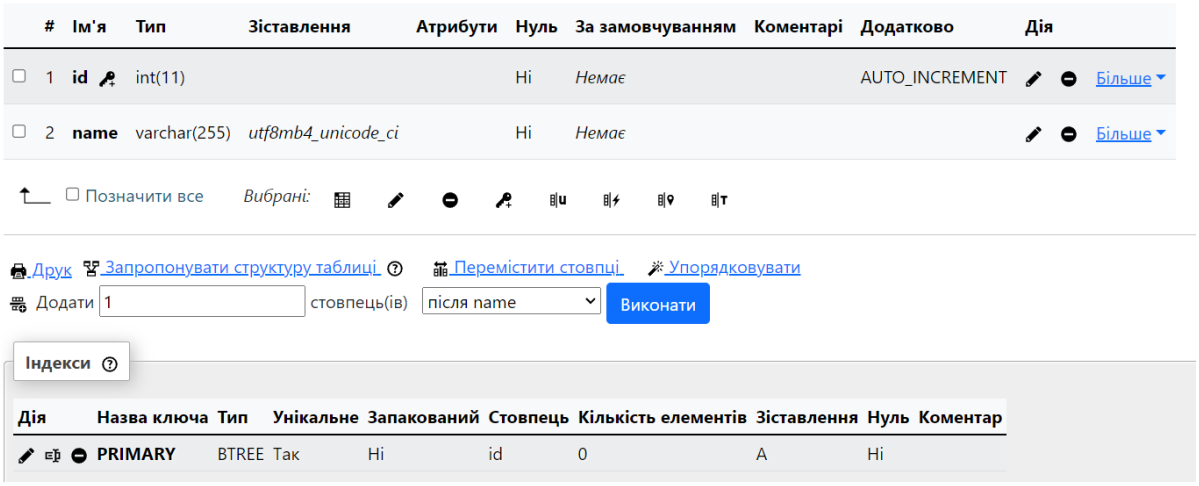


Рисунок 3.5 – Структура таблиці OptionsType

В таблиці **CategoryCodeType** містяться різні типи категорій в меню. Кожен тип має унікальний ідентифікатор (id) та назву (name). Структура таблиці відображена на рисунку 3.6.

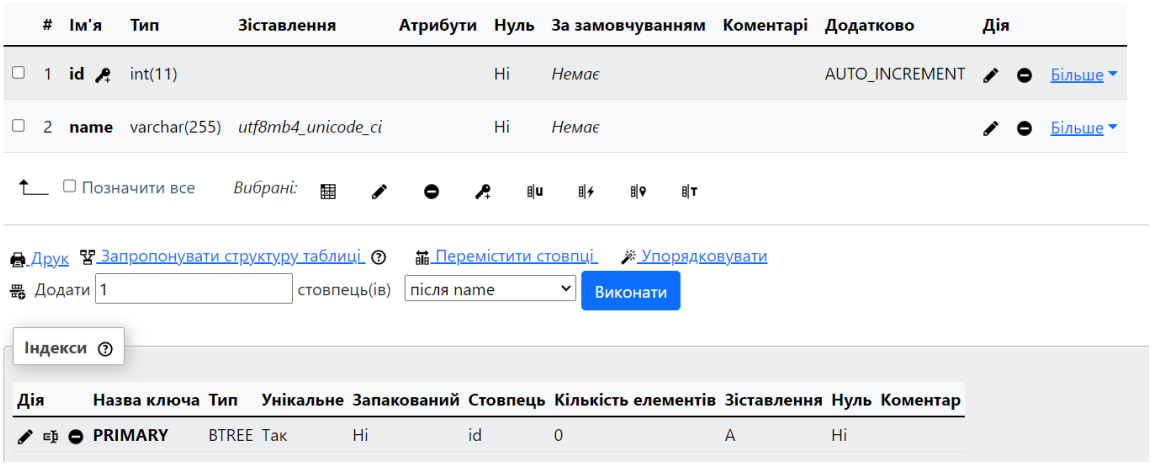
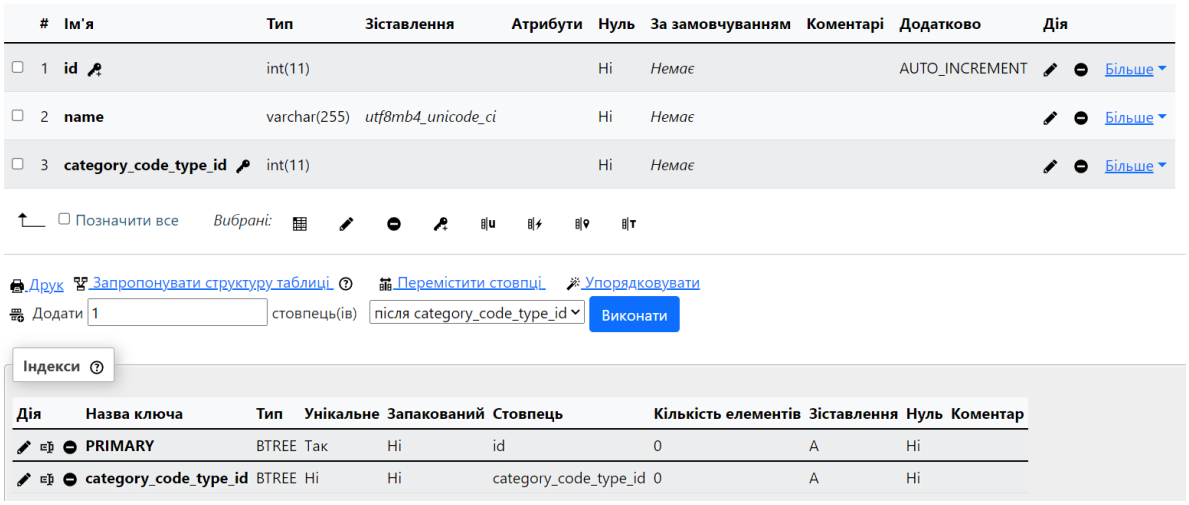


Рисунок 3.6 – Структура таблиці CategoryCodeType

Таблиця **CategoryType** представляє категорії певних позицій у меню. Кожна категорія має унікальний ідентифікатор (id), назву (name) та зв'язок з типом коду категорій (category_code_type_id). Структура таблиці представлена на рисунку 3.7.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	category_code_type_id	int(11)			Ні	Немає			Більше

Позначити все Вибрані:

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

Додати 1 стовпець(ів) після category_code_type_id Виконати

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	category_code_type_id	BTREE	Ні	Ні	category_code_type_id	0	A	Ні	

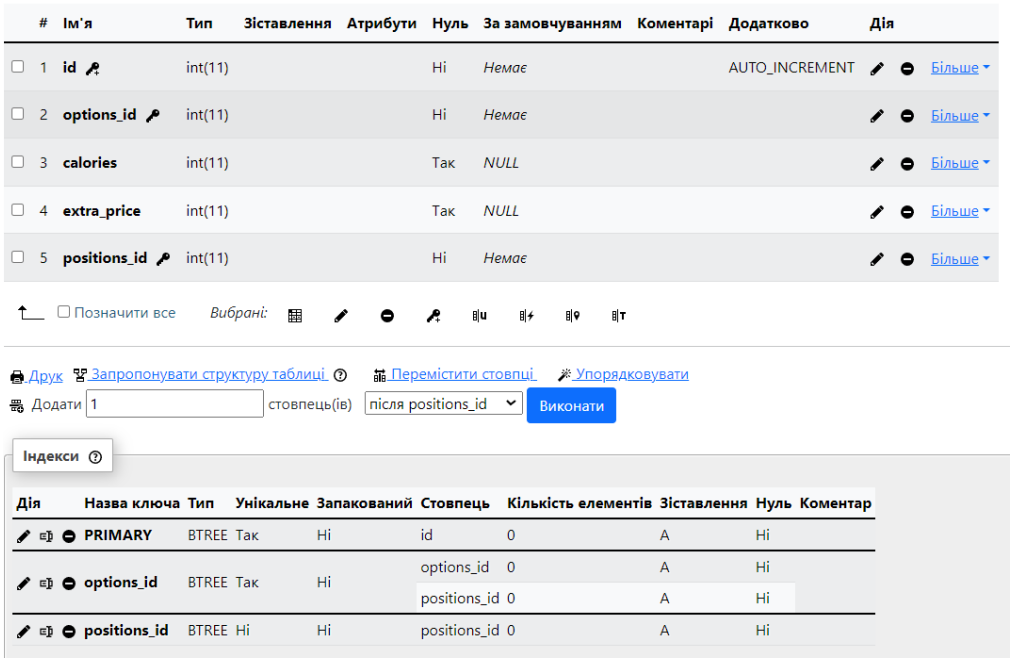
Рисунок 3.7 – Структура таблиці CategoryType

Таблиця **Positons** таблиця містить інформацію про певні позиції у меню. Кожна позиція має унікальний ідентифікатор (id), назву (name), зв'язок з типом категорії (category_type_id), кількість лайків (likes), ціну (price) та поточну знижку (current_discount). Структура таблиці представлена на рисунку 3.8.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
3	category_type_id	int(11)			Ні	Немає			Більше
4	likes	int(11)			Так	0			Більше
5	price	decimal(10,2)			Ні	Немає			Більше
6	current_discount	decimal(10,2)			Так	0.00			Більше

Рисунок 3.8 – Структура таблиці Positions

Таблиця **PositionsHasOptions** зберігає інформацію про опції, доступні для певних страв у меню. Кожний зв'язок між позицією та опцією має унікальний ідентифікатор (id), вказання на опцію (options_id), кількість калорій (calories), додаткову ціну (extra_price) та посилання на позицію (positions_id). Структура таблиці зображено на рисунку 3.9.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	options_id	int(11)			Ні	Немає			Більше
3	calories	int(11)			Так	NULL			Більше
4	extra_price	int(11)			Так	NULL			Більше
5	positions_id	int(11)			Ні	Немає			Більше

Позначити все Вибрані:

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

Додати 1 стовпець(ів) після positions_id Виконати

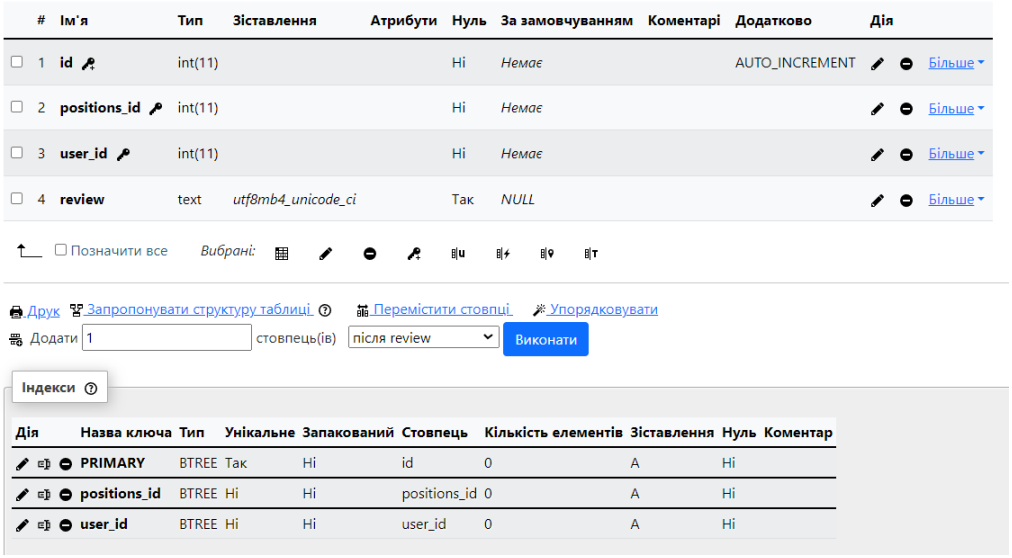
Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	options_id	BTREE	Так	Ні	options_id	0	A	Ні	
					positions_id	0	A	Ні	
	positions_id	BTREE	Ні	Ні	positions_id	0	A	Ні	

Рисунок 3.9 – Структура таблиці PositionsHasOptions

Таблиця **User** містить дані користувачів системи. Кожен користувач має унікальний ідентифікатор (id), ім'я (first_name), прізвище (last_name), телефон (phone), електронну адресу (email) та пароль (password).

В таблиці **PositionsReviews** зберігаються відгуки користувачів про певні страви чи напої в меню. Кожен відгук має унікальний ідентифікатор (id), посилання на позицію (positions_id), ідентифікатор користувача (user_id) та текст самого відгуку (review). Структура таблиці зображено на рисунку 3.10.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	Немає		AUTO_INCREMENT	Більше
2	positions_id	int(11)			Ні	Немає			Більше
3	user_id	int(11)			Ні	Немає			Більше
4	review	text	utf8mb4_unicode_ci		Так	NULL			Більше

Позначити все Вибрані:

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

Додати 1 стовпець(ів) після review Виконати

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	positions_id	BTREE	Ні	Ні	positions_id	0	A	Ні	
	user_id	BTREE	Ні	Ні	user_id	0	A	Ні	

Рисунок 3.10 – Структура таблиці PositionsReviews

Таблиця **Order** містить всю інформацію про замовлення. Кожне замовлення має унікальний ідентифікатор (id), посилання на користувача-замовника (user_id), дату (order_date), тип доставки (delivery_type), тип оплати (payment_type), кінцеву ціну (final_price), знижку (discount_price), промокод (promo) та коментар (comment). Структура таблиці представлена на рисунку 3.11.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	**id**	int(11)			Ні	*Немає*		AUTO_INCREMENT	Більше
2	**user_id**	int(11)			Ні	*Немає*			Більше
3	**order_date**	timestamp			Ні	current_timestamp()			Більше
4	**delivery_type**	int(11)			Ні	*Немає*			Більше
5	**payment_type**	int(11)			Ні	*Немає*			Більше
6	**final_price**	decimal(10,2)			Ні	*Немає*			Більше
7	**discount_price**	decimal(10,2)			Так	0.00			Більше
8	**promo**	varchar(255)	*utf8mb4_unicode_ci*		Так	*NULL*			Більше
9	**comment**	text	*utf8mb4_unicode_ci*		Так	*NULL*			Більше

Рисунок 3.11 – Структура таблиці Order

Таблиця **DeliveryType** містить різні типи доставки. Кожен тип має унікальний ідентифікатор (id) та назву (name).

Таблиця **DeliveryDestination** для зберігання інформації про адресу доставки для кожного замовлення. Кожна адреса має унікальний ідентифікатор (id), назву (name) та посилання на замовлення (order_id). Структура таблиці зображена на рисунку 3.12.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	**id**	int(11)			Ні	*Немає*		AUTO_INCREMENT	Більше
2	**name**	varchar(255)	*utf8mb4_unicode_ci*		Ні	*Немає*			Більше
3	**order_id**	int(11)			Ні	*Немає*			Більше

Позначити все *Вибрані:*

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

Додати 1 стовпець(ів) після order_id Виконати

Індекси

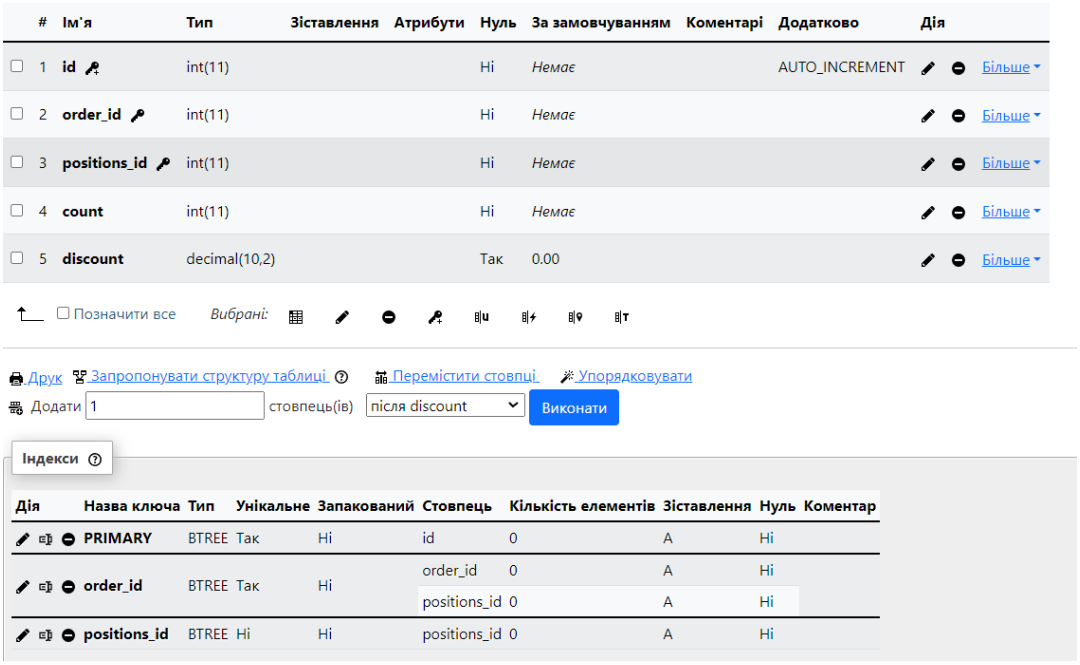
Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	order_id	BTREE	Ні	Ні	order_id	0	A	Ні	

Рисунок 3.12 – Структура таблиці DeliveryDestination

Таблиця **PaymentType**

Таблиця, що містить різні типи оплати. Кожен тип має унікальний ідентифікатор (id) та назву (name).

Таблиця **OrderHasPositions** для зберігання інформації про позиції, які входять до кожного замовлення. Кожна зв'язка між замовленням та позицією має унікальний ідентифікатор (id), посилання на замовлення (order_id), посилання на позицію (positions_id), кількість (count) та знижку (discount). Структура таблиці відображена на рисунку 3.13.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	int(11)			Ні	*Немає*		AUTO_INCREMENT	Більше
2	order_id	int(11)			Ні	*Немає*			Більше
3	positions_id	int(11)			Ні	*Немає*			Більше
4	count	int(11)			Ні	*Немає*			Більше
5	discount	decimal(10,2)			Так	0.00			Більше

Позначити все *Вибрані:*

Друк Запропонувати структуру таблиці Перемістити стовпці Упорядковувати

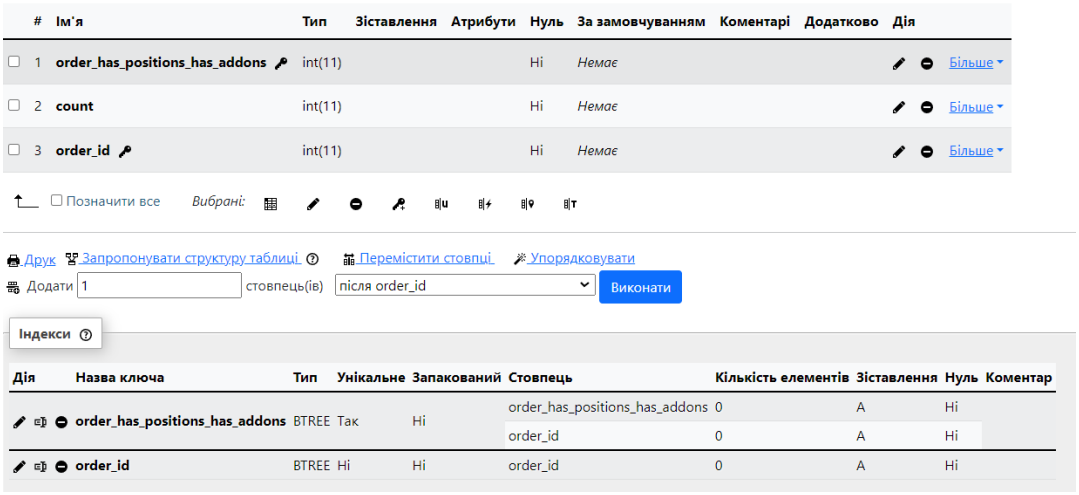
Додати 1 стовпець(ів) після discount Виконати

Індекси

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	PRIMARY	BTREE	Так	Ні	id	0	A	Ні	
	order_id	BTREE	Так	Ні	order_id	0	A	Ні	
					positions_id	0	A	Ні	
	positions_id	BTREE	Ні	Ні	positions_id	0	A	Ні	

Рисунок 3.13 – Структура таблиці OrderHasPositions

В таблиці **OrderHasPositionHasAddons** зберігається інформація про додаткові опції, вибрані для певних позицій у замовленні. Кожна зв'язка між позицією та опцією має кількість (count), унікальний ідентифікатор (order_has_positions_has_addons) та посилання на замовлення (order_id). Структура таблиці представлена на рисунку 3.14.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	order_has_positions_has_addons	int(11)			Ні	Немає			Більше
2	count	int(11)			Ні	Немає			Більше
3	order_id	int(11)			Ні	Немає			Більше

Позначити все Вибрані:

Дія	Назва ключа	Тип	Унікальне	Запакований	Стовпець	Кількість елементів	Зіставлення	Нуль	Коментар
	order_has_positions_has_addons	BTREE	Так	Ні	order_has_positions_has_addons	0	A	Ні	
					order_id	0	A	Ні	
	order_id	BTREE	Ні	Ні	order_id	0	A	Ні	

Рисунок 3.14 – Структура таблиці OrderHasPositionHasAddons

Ця таблиця є важливою задля індивідуального підходу в обробці замовлень, дозволяючи враховувати всі додаткові опції, які обирають клієнти.

Висновки до розділу 3

У третьому розділі КРБ детально проаналізовано стек технологій для розробки проєкту. Ще описано технології та інструменти – це вибір мови програмування, фреймворків та бібліотек, які найбільше підходять для досягнення поставленої мети. Також при розробці моделі БД були виділені вимоги та обмеження даних, та створено логічну модель, що забезпечує надійне зберігання та ефективне управління даними.

Отож, цей розділ присвячений етапам розробки проєкту, починаючи з вибору технологій та завершуючи створенням надійної та оптимізованої БД, що сприяє результативному досягненню поставленої мети.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

Вебзастосунок для кафе навинос має просту структуру та зрозумілу навігацію, в хедері сайту є основні сторінки, кошик та можливість авторизуватись. Після входу в систему, також є можливість отримувати повідомлення про зміну статусу замовлення. У футері сайту додатково зазначено ще контакти. Загалом, можна було б додати можливість в режимі гостя замовляти, але щодо авторизації є декілька переваг: зберігання даних та історії замовлень, бонусна програма, зворотній зв’язок, що допомагає покращити якість обслуговування.

На сайті можна авторизуватись як користувач, та як адміністратор, для другого є окрема панель для роботи та обробки замовлень, відгуків та перегляду звітності. Враховуючи поставлені завдання, додана можливість виконання CRUD-операцій з замовленнями та відгуками. Основна мета сайту – швидко замовити їжу та напої з доставкою або самовивозом.

4.1 Реалізація вебсайту

Для реєстрації потрібно небагато даних, а саме це: ім’я, телефон, пошта, пароль та підтвердження. А вхід здійснюється через номер та пароль. Реалізація сторінки на рисунку 4.1.

У разі потреби відновити пароль, то це здійснюється через введення пошти, відображення сторінки на рисунку 4.2.

Заповнений профіль користувача є на рисунку 4.3. В кошику клієнт може видаляти позиції та змінювати їх кількість. Після підтвердження замовлення з’являється модальне вікна з номером замовлення та способом доставки.

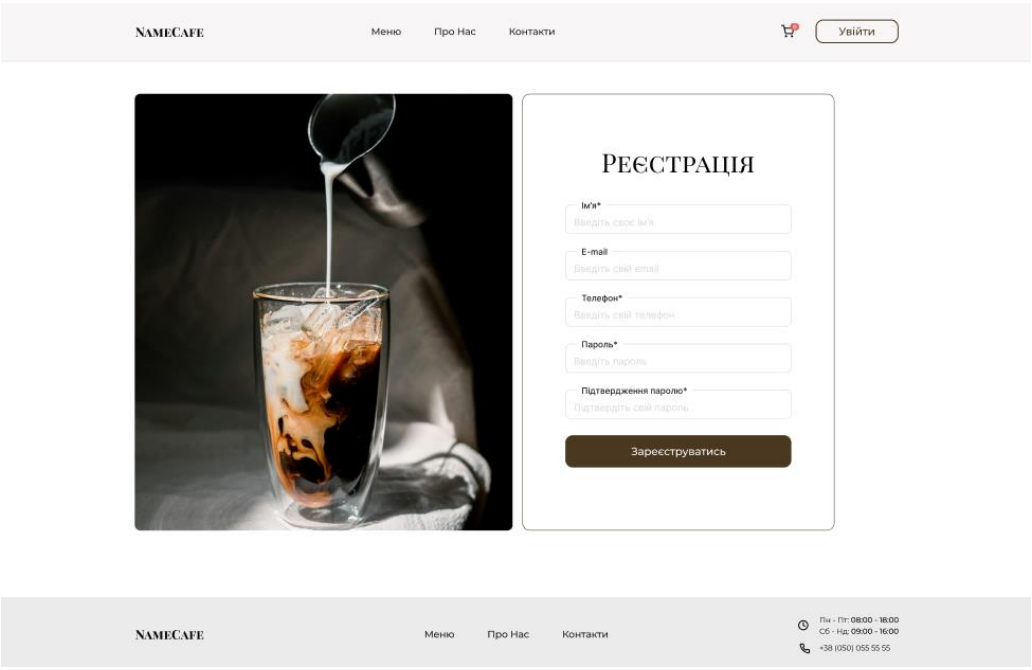


Рисунок 4.1 – Сторінка реєстрації

На формі реєстрації необхідно ввести всі дані, але необов’язковим полем є електронна пошта, також для полів задані правила валідації, перевірка правильності пошти та збіг паролів.

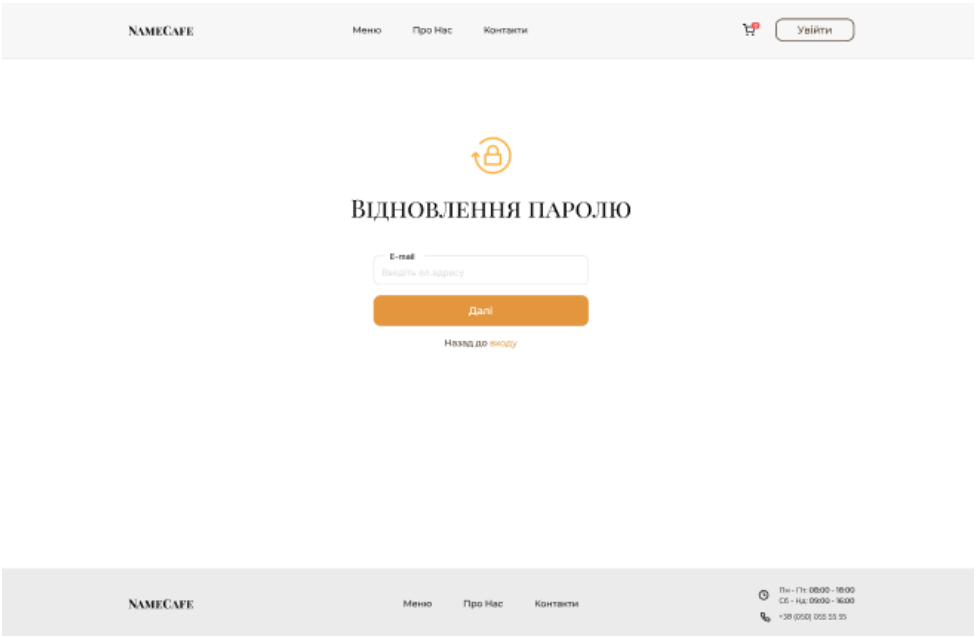


Рисунок 4.2 – Сторінка відновлення паролю

Модальне вікно, що відображено на рисунку 4.4 є прикладом широкого вибору опції для напою, також там відразу зазначена ціна, та є можливість додати цей напій в «Улюблене» розділ.

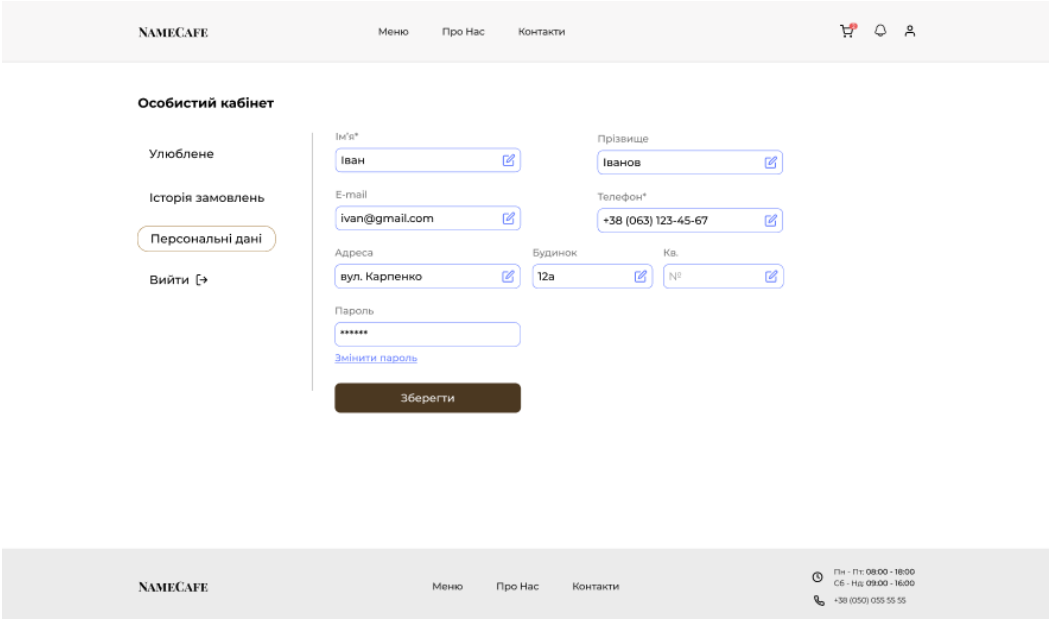


Рисунок 4.3 – Сторінка профілю

Відгуки відображені будуть при натисканні на однойменний розділ.

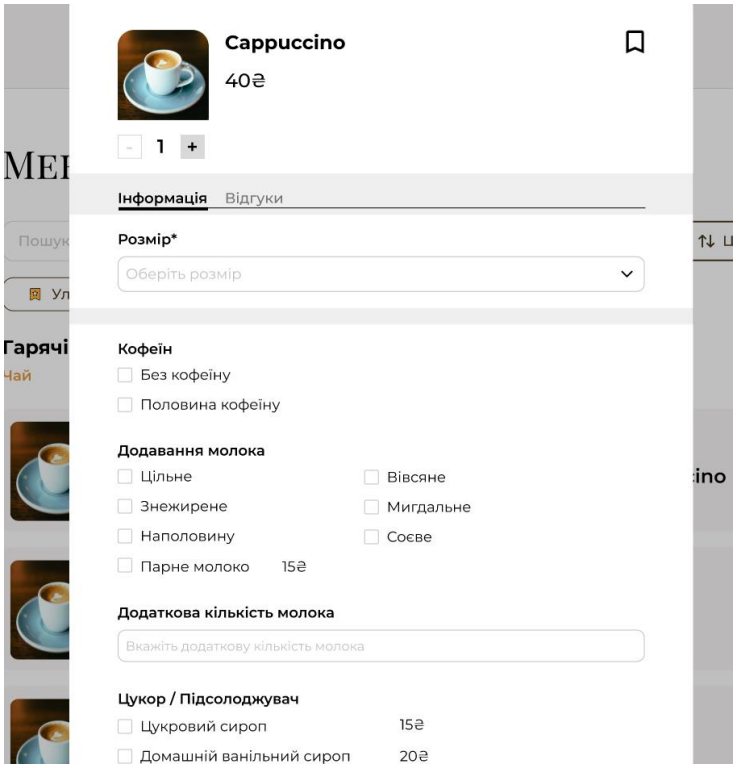


Рисунок 4.4 – Модальне вікно

Наприклад, в категорії напої в карточці товару більшість має опції, такі як вибір розміру, додавання додатково сиропу або молока, а також за потреби, можна дописати індивідуальні вимоги в опції «Спеціальні запити».

4.2 Реалізація панелі адміністрування

Для зручного адміністрування вебзастосунку розроблено панель адміністратора. Це важливий інструмент для управління бізнесом та має безліч функціоналу. Адміністратор може редагувати меню: додавати, змінювати та видаляти позиції. Також це відстеження та оновлення статусу замовлення, ще це відстеження продажі, прибутків та кількість клієнтів. Важливо відстежувати діяльність клієнтів, а саме відгуки, тому також є можливість видалити їх. Таблиця з поточними замовленнями можна переглянути на рисунку 4.5. У колонці «Дія» можна перегляну та відразу внести зміни до нього або ж видалити. Перед видаленням з’являється модальне вікно підтвердження.

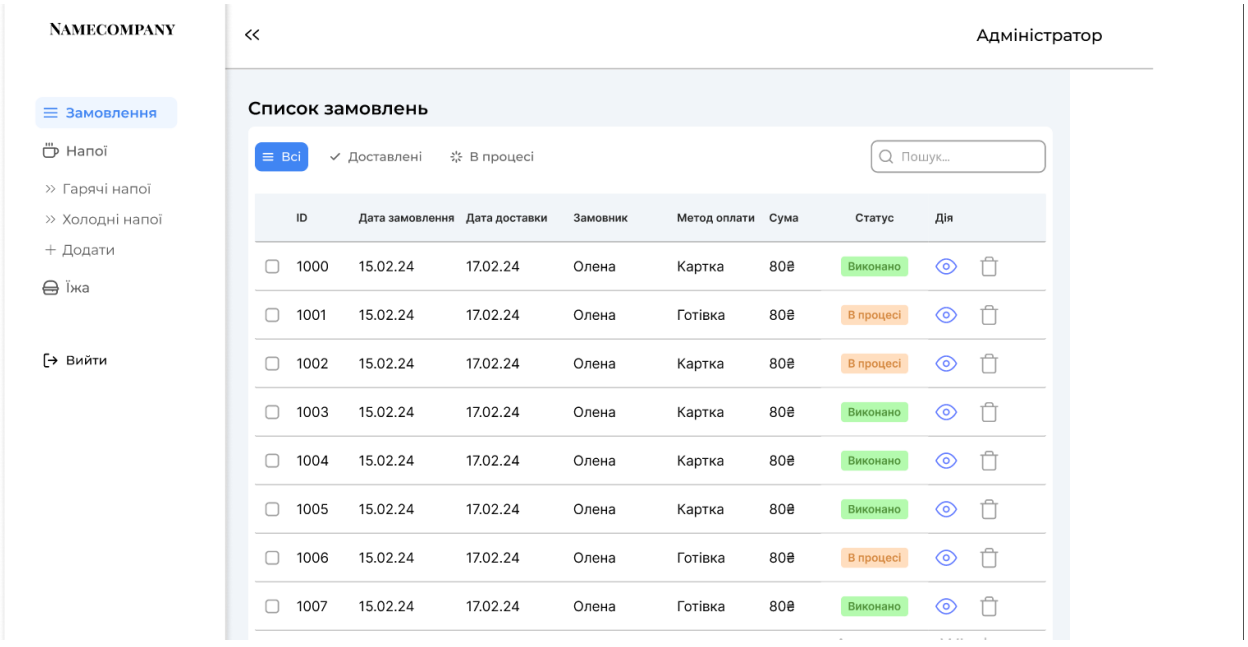


Рисунок 4.5 – Таблиця замовлень

Щодо створення нової позиції, то тут багато полів для заповнення адже є можливість редагування як основних даних, так й специфічних, наприклад це задання розмірів напої та опції додаткові (сиропи, цукор, вибір молока та інше).

Усе можна налаштовувати індивідуально під кожну позицію. Також є завантаження картинки, результат відображено на рисунку 4.6. Обов'язковими полями є розділ даних продукту та зазначення розміру або порції.

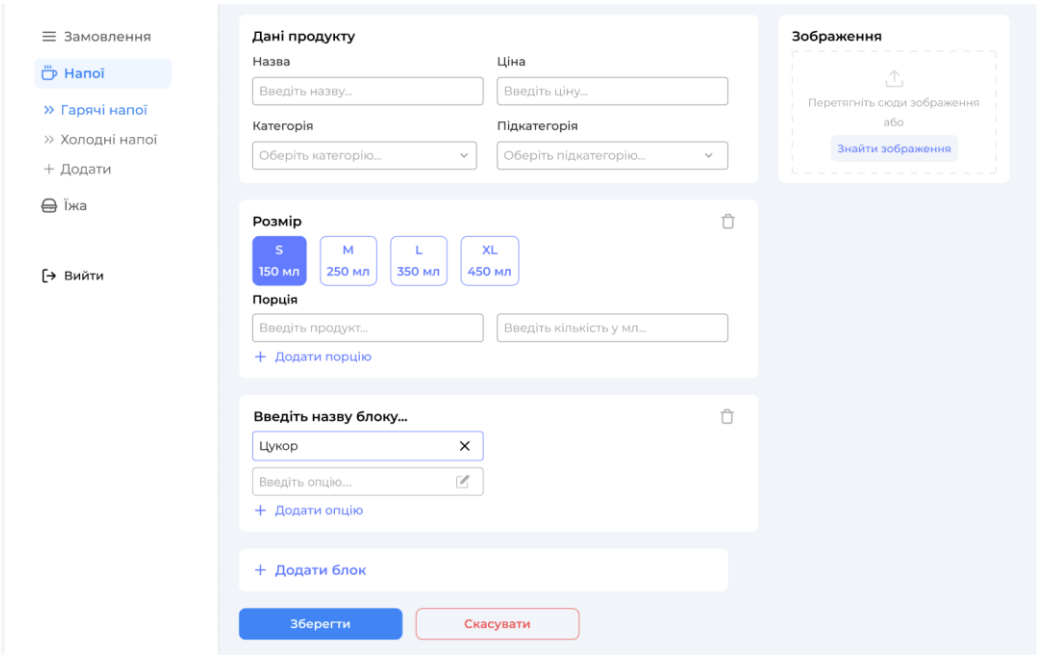


Рисунок 4.6 – Створення нової позиції

Блок про основну інформацію: замовника, складові замовлення з примітками та розписано дані про оплату, все це відображено на рисунку 4.7.

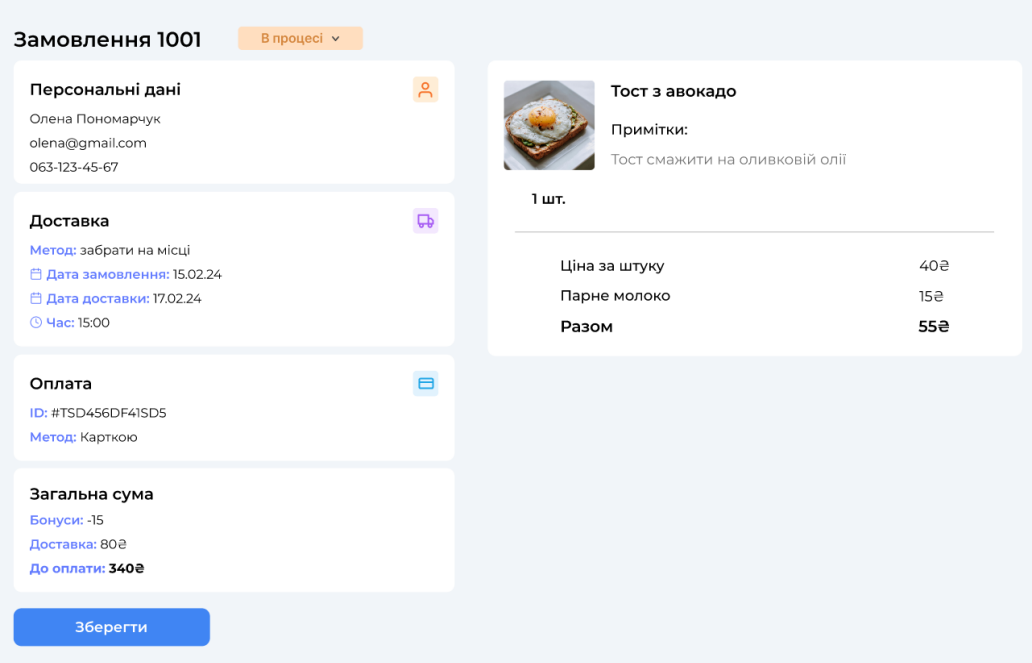


Рисунок 4.7 – Інформація про замовлення

У створеному замовленні є можливість лише змінити статус, замовник автоматично отримає повідомлення.

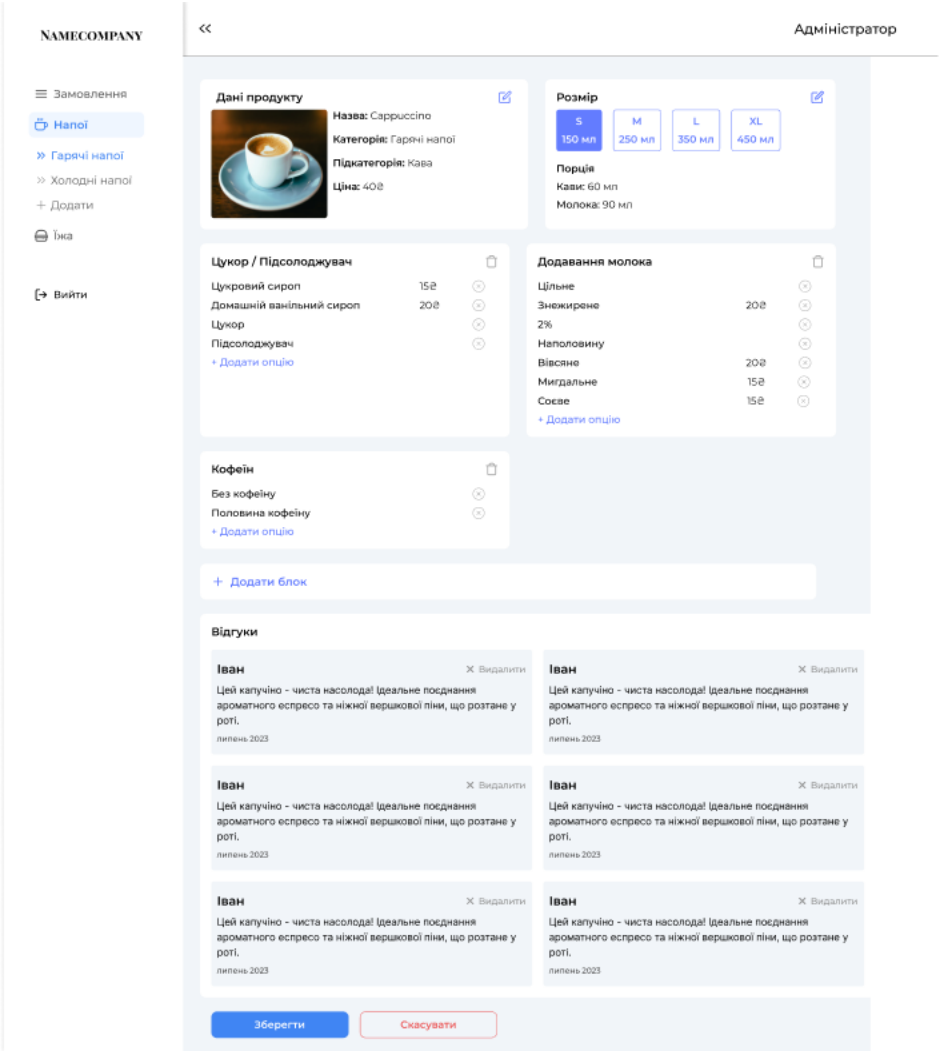


Рисунок 4.8 – Сторінка адміністрування певного товару

Сторінка адміністрування певного напоя представлена на рисунку 4.8: можна видалити опції в підрозділах, також змінити порції в розмірах та додати новий блок за потребою. Також є перегляд усіх відгуків.

4.3 Реалізація компонентів логіки застосунку

Реалізація системи керування замовленнями дозволить клієнтам замовляти онлайн, а Laravel забезпечує зручний спосіб розробки для обміну даними між клієнтами і сервером [17]. При розробці сайту для кафе навинос, використання фреймворку значно спрощує роботу з моделями та контролерами, а також із базою

даних. Зручна та гнучка архітектура, на шаблоні MVC робить код організованим. В реалізації сайту кафе важливо правильно організувати роботу з БД, щоб забезпечити ефективність, безпеку та надійність.

Моделі даних необхідні для відображення даних та роботи з БД, це потрібно для CRUD-операції з даними. *Контролери* потрібні для обробки запитів та виклик відповідних методів моделей. Завдяки цьому можна відокремити бізнес-логіку від представлення даних.

Першим прикладом буде контролер для користувачів, який зображено на рисунку 4.9: він спрямований на такий функціонал як автентифікація та авторизація, управління обліковими записами користувачів, ролі та права доступу. А саму модель користувачів відображено на рисунку 4.10.

Опис методів:

index()

- отримання всіх користувачів з БД за допомогою User::all();
- повернення представлення users.index, передаючи список користувачів.

create()

- повернення представлення users.create, яке відображає форму для створення нового користувача.

store(Request $request)

- валідація вхідних даних запиту, забезпечуючи, що ім'я, електронна пошта та пароль відповідають певним правилам;
- хешування паролю за допомогою bcrypt;
- створення нового користувача у БД з використанням методу User::create;
- перенаправлення на список користувачів з повідомленням про успішне створення користувача.

show($id)

- пошук користувача за ідентифікатором $id або повертається помилка 404, якщо користувача не знайдено;

– повернення представлення users.show, передаючи знайденого користувача.

```
class UserController extends Controller
{
    public function index()
    {
        $users = User::all();
        return view('users.index', compact('users'));
    }

    public function create()
    {
        return view('users.create');
    }

    public function store(Request $request)
    {
        $validated = $request->validate([
            'name' => 'required|string|max:255',
            'email' => 'required|email|unique:users,email',
            'password' => 'required|string|min:8',
        ]);

        $validated['password'] = bcrypt($validated['password']);

        User::create($validated);

        return redirect()->route('users.index')->with('success', 'User
created successfully.');
    }

    public function show($id)
    {
        $user = User::findOrFail($id);
        return view('users.show', compact('user'));
    }
```

Рисунок 4.9 – Контролер для обробки користувачів

edit($id)

– пошук користувача за ідентифікатором $id або повертає помилку 404;

– повернення представлення users.edit, передаючи знайденого користувача.

update(Request $request, $id)

– валідація вхідні дані запиту. Електронна пошта перевіряється на унікальність;

– якщо пароль присутній у запиті, він хешується;

– пошук користувача за ідентифікатором $id або повертає помилку 404;

– оновлення даних користувача в БД;

– перенаправлення на список користувачів з повідомленням про успішне оновлення користувача.

destroy($id)

– пошук користувача за ідентифікатором $id або повертає помилку 404;

– видалення користувача з БД;

– перенаправлення на список користувачів з повідомленням про успішне видалення користувача.

```php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    use HasFactory;

    protected $table = 'user';

    protected $fillable = [
        'first_name', 'last_name', 'phone', 'email', 'password'
    ];

    public $timestamps = false;

    public function orders()
    {
        return $this->hasMany(Order::class, 'user_id');
    }

    public function reviews()
    {
        return $this->hasMany(PositionsReviews::class, 'user_id');
    }
}
```

Рисунок 4.10 – Модель користувачів

Наступним буде описано контролер для категорії продуктів: він використовує моделі CategoryType та CategoryCodeType, щоб надати функціональність створення, читання, оновлення та видалення типів категорій. Детальний опис декілька методів:

index()

– отримання всіх типів категорій разом з пов'язаними типами кодів категорій (categoryCodeType) за допомогою CategoryType::with('categoryCodeType')->get();

– повернення представлення category_types.index, передаючи список типів категорій.

create()

– отримання всіх типів кодів категорій (CategoryCodeType::all());

– повернення представлення category_types.create, передаючи список типів кодів категорій для вибору, як відображено на рисунку 4.11.

```
    public function create()
    {
        $categoryCodeTypes = CategoryCodeType::all();
        return view('category_types.create', compact('categoryCodeTypes'));
    }

    public function store(Request $request)
    {
        $validated = $request->validate([
            'name' => 'required|string|max:255',
            'category_code_type_id' =>
'required|exists:category_code_type,id',
        ]);

        CategoryType::create($validated);

        return redirect()->route('category_types.index')->with('success',
'Category Type created successfully.');
    }
```

Рисунок 4.11 – Метод create() для категорій

store(Request $request)

– валідація вхідних даних запиту, забезпечуючи, що поле name є обов'язковим та має максимальну довжину 255 символів, а поле category_code_type_id існує у таблиці category_code_type;

– створення нового типу категорії у БД за допомогою методу CategoryType::create;

– перенаправлення на сторінку зі списком типів категорій з повідомленням про успішне створення.

edit($id)

– пошук типу категорії за ідентифікатором $id або повертає помилку 404;

– отримання всіх типів кодів категорій (CategoryCodeType::all());

– повернення представлення category_types.edit, передаючи знайдений тип категорії та список типів кодів категорій для вибору.

destroy($id)

– пошук типів категорії за ідентифікатором $id або повертає помилку 404, якщо тип категорії не знайдено;

– видалення типу категорії;

– перенаправлення на сторінку зі списком типів категорій з повідомленням про успішне видалення.

Контролер для замовлень, з використанням моделі, що на рисунку 4.12, взаємодіє з моделями Order, User, DeliveryType, та PaymentType, надаючи функціональність для CRUD-операцій, читання, оновлення та видалення замовлень. Також він забезпечує взаємодію між клієнтами, адміністраторами, та БД, опис методів:

index()

– отримання всіх замовлення разом з пов'язаними користувачами, типами доставки та типами оплати за допомогою Order::with(['user', 'deliveryType', 'paymentType'])->get();

– повернення представлення orders.index, передаючи список замовлень.

```
class Order extends Model
{
    use HasFactory;

    protected $table = 'order';

    protected $fillable = [
        'user_id', 'order_date', 'delivery_type', 'payment_type',
'final_price',
        'discount_price', 'promo', 'comment'
    ];

    public $timestamps = false;

    public function user()
    {
        return $this->belongsTo(User::class, 'user_id');
    }

    public function deliveryType()
    {
        return $this->belongsTo(DeliveryType::class, 'delivery_type');
    }

    public function paymentType()
    {
        return $this->belongsTo(PaymentType::class, 'payment_type');
    }

    public function deliveryDestination()
    {
        return $this->hasOne(DeliveryDestination::class, 'order_id');
    }

    public function positions()
    {
        return $this->belongsToMany(Positions::class,
'order_has_positions', 'order_id', 'positions_id')
            ->withPivot('count', 'discount');
    }
```

Рисунок 4.12 – Модель замовлень

create()

– отримання всіх користувачів за допомогою User::all();

– отримання всіх типів доставки за допомогою DeliveryType::all();

– отримання всіх типів оплати за допомогою PaymentType::all();

– повернення представлення orders.create, передаючи списки користувачів, типів доставки та типів оплати.

update(Request $request, $id)

– валідація вхідних даних запиту. Перевірка, що всі необхідні поля відповідають певним правилам;

– пошук замовлення за ідентифікатором $id або повертає помилку 404;

– оновлення даних замовлення у базі даних;

4.4 Тестування розробленого ПЗ

Тестування програмного забезпечення – це процес виконання програми або системи з метою виявлення помилок. Це передбачає будь-яку діяльність, спрямовану на оцінку програми або системи та визначення того, що вона відповідає необхідним результатам. Але важливо зазначити, що також вірно, що для будь-якої системи, дефекти при конструюванні та розробці ніколи не можна повністю виключити [18].

Тестування ПЗ для кафе має важливу роль у забезпеченні якості та ефективності, адже клієнти очікують постійного доступу та гарного обслуговування, обробка платежів та захист інформації необхідні для кожного ПЗ, та також виявлення помилок на етапі тестування буде коштувати менше організації.

Можна виділити основні принципи тестування [19]:

– тестування може показати наявність дефектів, але не довести їх відсутність, важливо складати тест-кейси;

– вибір методології, техніки та типу тестування безпосередньо залежить від вимог та функціоналу самого застосунку;

– тести повинні бути незалежними один від одного, щоб уникнути взаємного впливу та забезпечити об'єктивні результати;

– постійне повторювання.

Таблиця 4.1 – Процес створення нового блюда в меню

Діючі особи	Адміністратор
Мета	Додати до меню нову страву
Передумова	Працівник авторизувався як адміністратор
Успішний сценарій: 1) Адміністратор авторизується та входить до адмін панелі 2) В адмін панелі адміністратор переходить в розділ «Їжа» та натискає на кнопку «Додати страву». 3) Система відображає форму для введення інформації про нове блюдо: назву, ціну, категорію, склад, калорійність, порцію, завантаження зображення. 4) Адміністратор вводить всю необхідну інформацію про нове блюдо. 5) Адміністратор завантажує фото страви (за потреби). 6) Після заповнення всіх полів адміністратор натискає кнопку "Зберегти". 7) Система перевіряє введені дані та зберігає нове блюдо у БД кафе. 8) Система повідомляє адміністратора про успішне додавання нового блюда. 9) Адміністратор може переглянути нове блюдо в меню на сайті або в панелі.	
Сценарій успішний. Нова страва додана до меню.	
Розширення:	
1a	Адміністратор не заповнив всі обов'язкові поля або введена інформація не відповідає валідації, система відображає модальне вікно про помилку та підсвічує невірно заповнені поля. Результат: нова страва не додана в меню.
1b	Створення нової страви з існуючою назвою, у такому випадку система також виводить повідомлення про помилки. Результат: зміни не зберігаються.
Усі сценарії розширення успішно виконані.	

Таблиця 4.2 – Додавання відгуку до страви або напоя

Діючі особи	Користувач
Мета	Залишити відгук до позиції в меню
Передумова	Користувач повинен бути авторизованим в системі
Успішний сценарій: 1) Користувач вводить правильні дані та авторизується в системі. 2) Переходить на сторінку «Меню» та шукає необхідну позицію. 3) Після цього натискає на неї та в карточці товару переходить до розділу «Відгуки». 4) Заповнює форму та натискає на кнопку «Надіслати». 5) Система виводить повідомлення про успішне додавання відгуку. 6) Користувач може переглянути його на сайті.	
Сценарій успішний. Відгук опублікований на сайті.	
Розширення:	
1a	Після публікації відгуку, користувач не може редагувати його. Результат: видалення відгуку може зробити тільки адміністратор.
1b	Користувач не зареєстрований, переходить до розділу з відгуками, але на карточці товару немає відповідної кнопки «Написати відгук». Результат: користувач може лише переглянути чужі відгуки.
Усі сценарії розширення успішно виконані.	

Таблиця 4.3 – Змінення даних в особистому кабінеті

Діючі особи	Користувач
Мета	Оновити дані в кабінеті
Передумова	– Користувач зареєстрований і увійшов у систему. – Користувач знаходиться на сторінці особистого кабінету.
Успішний сценарій: 1) Користувач переходить в розділ «Персональні дані». 2) Для редагування потрібно натиснути на необхідне поле. 3) Змінює дані. 4) Після цього він може змінити інші поля за необхідністю, або залишити лише одну зміну. 5) Натискає на кнопку «Зберегти».	

Кінець таблиці 4.3

Сценарій успішний. Виводиться повідомлення про успішне збереження. Зміни у полях відображаються правильно.	
Розширення:	
1а	При введенні некоректних даних у будь-яке поле, з'являється відповідне повідомлення про помилку. Зміни не зберігаються, доки всі дані не будуть введені правильно.
Усі сценарії розширення успішно виконані.	

Ручне тестування залишається важливим методом забезпечення якості та надійності функціоналу ПЗ, наприклад як у випадку сфери харчування. Воно дозволяє активно взаємодіяти з системою як потенційні користувачі. Певним недоліком є те, що цей вид тестування потребує великих часових затрат та людського ресурсу, при цьому не виключається людський фактор, а саме неуважність [20]. Гарною перевагою можна виділити те, що завдяки такому тестування можна відтворити реальні сценарії використання, такі як замовлення страв, реєстрація, платежі та залишити відгук. Це точно дає повну відповідь, що система працює належним чином.

Висновки до розділу 4

У четвертому розділі КРБ детально розглянуто реалізацію ПЗ. Описано процес використання панелі адміністрування, що є важливим компонентом для керування меню, замовлень. Також особливу увагу приділено реалізації контролерів та моделей, що є ключовою частиною архітектури MVC. Це допомагає зрозуміти логіку та взаємодію компонентів. Продемонстровано приклади коду щодо принципів роботи ПЗ.

Також проведено тестування, що засвідчило коректність взаємодії з БД, цілісність даних та правильність реалізації бізнес-логіки, що є підтвердженням надійності функціоналу та досягнення поставленої мети проєкту.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра покращено ефективність роботи та автоматизовано систему ведення замовлень, страв та клієнтської бази кафе навинос шляхом розробки вебзастосунку.

Для досягнення визначеної мети вирішено такі завдання:

– проаналізовано предметну область, аналоги та визначено вплив сучасних технологій;
– визначено вимоги до ПЗ та функціонал усіх сторінок застосунку;
– складено сценарії використання;
– розроблено дизайн;
– спроєктовано модель бази даних меню, користувачів;
– розроблено та реалізовану логіку застосунку;
– протестовано систему.

Визначення специфікації вимог до вебзастосунку забезпечило чітке розуміння функціоналу, необхідного для задоволення потреб користувачів. Сценарії використання відображають основні взаємодії користувачів із системою. Описано стек технологій розробки, зокрема використання Tailwind CSS, Laravel та MySQL. Завдяки розробці діаграми класів та моделі бази даних, забезпечено структуровану архітектуру програмного забезпечення.

Реалізовано сайт із основним функціоналом для користувачів, наприклад перегляд меню, реєстрація, оформлення замовлення, додавання улюблених страв. Створено адміністративну панель для управління замовленнями та меню. Проведено ручне тестування для надійності системи. Результатом проведеної роботи є вебзастосунок для кафе навинос.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1) Діджиталізація бізнесу: сьогодення і майбутнє. Збірник матеріалів круглого столу (м. Київ, 28 січня 2021 року). Київ: «Хай-Тек Прес», 2021. 88 с.

2) Онлайн-ресторан «Вилки-Палки». URL: https://vilki-palki.od.ua (Дата звернення: 22.04.2024)

3) Online їдальня: доставка їжі «SimpleFood». URL: https://simple.mk.ua (Дата звернення: 22.04.2024)

4) Онлайн-ресторан «NaDivanE». URL: https://nadivaneda.com.ua (Дата звернення: 22.04.2024)

5) Bell D. UML basics: The sequence diagram. *Retrieved July*. Vol. 17, 2004. P. 14. 2015.

6) Єфремов М. Ф., Єфремов Ю. М., Єфремов В. М. Проектування програмного забезпечення з використанням UML. С.2. 2016.

7) Rivero J. M., Rossi G., Grigera J. et al. From mockups to user interface models: an extensible model driven approach. *Current Trends in Web Engineering: 10th International Conference on Web Engineering ICWE 2010 Workshops, Vienna, Austria, July 2010, Revised Selected Papers 10*(2010). P. 13–24.

8) Костенко О. Б. Організація баз даних та знань : конспект лекцій / О. Б. Костенко, І. О. Гавриленко; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. Харків: ХНУМГ ім. О. М. Бекетова, 2021. 92 с.

9) Stauffer M. Laravel: Up & Running. «O'Reilly Media, Inc.», 2023. 519 P.

10) Laaziri M., Benmoussa K., Khoulji S. et al. A Comparative study of PHP frameworks performance. *Procedia Manufacturing*. Vol. 32, 2019. P. 864–871. DOI:https://doi.org/10.1016/j.promfg.2019.02.295.

11) Sinha S., Dave H. J. Beginning Laravel. Springer, 2017. P. 189.

12) Gilmore W. Beginning PHP and MySQL: from novice to professional. Apress; 2010. P. 922. URL: https://salapura.com/www/InternetProgramiranje/Beginning-PHP-and-MySQL-5-2nd-Edition.pdf (Last acessed: 03.06.2024).

13) Tailwind CSS tutorial. URL: https://tw-elements.com/learn/te-foundations/tailwind-css/about/ (Last acessed: 04.06.2024).

14) Підручник з Umbrello UML Modeller. URL: https://docs.kde.org/trunk5/uk/umbrello/umbrello/ (Дата звернення: 26.04.2024)

15) Кодола Г. М., Волинець Н. С. Огляд інструментів для побудови UML-діаграм класів. ББК 32.81 С 91. 2013.

16) Golubinskyy R. Етапи проектування та моделювання баз даних. *Природничі та гуманітарні науки. Актуальні питання*. с. 40.

17) Sunardi A., others. MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based. *Procedia Computer Science*. Vol. 157, 2019. P. 134–141. URL: https://doi.org/10.1016/j.procs.2019.08.150 (Last accessed: 01.06.2024).

18) Pan J. Software testing. *Dependable Embedded Systems*. Vol. 5, Issue 2006. P. 1. 1999.

19) Принципи тестування. URL: https://qalight.ua/baza-znaniy/printsipi-testuvannya/ (Дата звернення: 09.06.2024)

20) Вовчарук П. Ю., Озеранський В. С., Перевозніков С. І. Особливості тестування фунціоналу вебзастосунків. ВНТУ, 2021.